Student Financial Assistance
Web Graphical User Interface (GUI) Guidelines

STATUS: DRAFT

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1. Document Purpose

The Web Graphical User Interface (GUI) Guidelines is intended to be read and used by all developers of SFA Web applications and Web pages.

To ease navigation and promote consistent Intranet and portal applications among all SFA Web sites and applications, all Web content should have a common look and feel. In other words, all content within a Web site, and to a certain degree, among all SFA Web sites, should share the same writing style, graphics, colors, and user interface elements.

These guidelines present the overall SFA Web philosophy and should guide the design process, as well as set forth coding, accessibility and usability standards for SFA Web site and page design. The guidelines promote completeness and a consistent interface among all SFA Web sites, but they also allow developers a certain degree of freedom and creativity, and the ability to tailor Web applications according to its primary audience.

This is a working document that combines the best of many similar HTML guides (see References). However, this version 1.9 is provided as a discussion draft document. Feedback and input to the GUI standards is requested and appreciated from all interested parties, especially developers and managers of SFA web applications and pages.

This document includes:

- Document naming conventions that illustrate the proper file-naming techniques.
- Site and page structure guidelines that provide guidelines on the usage of links, file sizes, and page appearance.
- Guidelines for the creation of Web content and multimedia files.
- HTML coding standards, accessibility, and usability standards.
- Examples of screen layout, navigation, headers, footers, and other design issues are provided in the form of generalized templates, which should be used as a model for developing and maintaining SFA Web pages.
- The Additional Resources section points to other Style Guides, HTML specifications, HTML primers, quick references, and other helpful resources.

All SFA Web sites and Web applications should adhere to SFA Web GUI Guidelines.

1.2. Requirements and Recommendations

This section presents items as either requirements or recommendations. A **Requirement** means that the specified item is an SFA standard. All GUI requirements must be followed in order for the Web page or Web site to be published. A "check" indicates that the specified item is a requirement, and the item usually contains the word "must."

For example:

√ Never use spaces in a file name. If a file name must be separated, use an underline to join the two words together (e.g. "file_name.html").

Recommendations are items that should be followed unless there is a sound business reason not to do so. Recommendations are indicated with triangles, and they usually use the word "should."

For example:

Δ You should provide alternatives to audio or video, such as a textual transcript. Contact the Webmaster (SFAWebmaster@ed.gov) with questions about Web style.

1.3. Exceptions

Developers may request an exception in order to override a specified recommendation or requirement by following the "Evergreening" Process outlined in the [PROVIDE SOURCE DOCUMENT].

1.4. User Interface Principles

A user-friendly interface makes it easy for users to tell the computer what they want to do, for the computer to request information from the users, and for the computer to present understandable information. Clear communication between the user and the computer is the working premise of good user interface design.

All user interface services at the SFA will be Section 508 compliant. Section 508 is a part of the Rehabilitation Act of 1973 which requires that electronic and information technology developed, procured, maintained, or used by the Federal government be accessible to people with disabilities New standards have been established to help Federal agencies

determine whether or not a technology product or system is accessible. Federal agencies must comply with these technology accessibility standards for all electronic and information technology acquired on or after six months from the date the Access Board issued its final standards. The final standards were published in the Federal Register on December 21, 2000. For more information on Section 508 compliance standards, see Section 2.3 of the Technology Handbook.

The user interface you create should accomplish the following:

- It must have enough visual appeal to attract a user to try the system for the first time.
- It must provide easy access to information so first-time users don't become intimidated.
- It should make the underlying technology disappear.
- Your interface and your information should reinforce the individual's understanding of the SFA community and their role as a member of it.
- The information you provide should be constantly refreshed and vibrant enough so users realize its long-term value and don't tire of seeing it – in fact, it should create increased use with time.
- It should keep the users and their goals in mind.
- The design approach should consider different levels of users experience (e.g. novice, experienced), user information, transaction needs, and frequency of use.

Good sites generally embody the well-established design and user interface principles identified in Elements of Good User Interfaces (see Table 1-1 below).

Table 1-1: Elements of Good User Interfaces

User Interface Requirement	Description
Clear	An understandable and intuitive user interface helps prevent user errors, makes important information obvious, and contributes to ease of learning and use.
Simple	The best interface designs are simple. Simple designs are easy to learn and to use and give the interface a consistent look. A good design requires a good balance between maximizing functionality and maintaining simplicity through progressive disclosure of information, logical layout and consistent framework.
Aesthetic	Every visual element that appears on the screen potentially competes for the user's attention. Provide an environment that is pleasant to work in and contributes to the user's understanding of the information presented. Give prominence to the most relevant info or content on the page according to user need and usage.
User-Controlled	The user, not the computer, initiates and controls all actions. Make it clear to users what effect their actions have.

Provides Immediate User Feedback	The interface should keep the user informed and provide immediate feedback. • User feedback is primarily visual. For example, using a client side script, e.g., JavaScript, to change the color of links as the mouse passes over them gives the user immediate feedback (audio can also be used to provide feedback, but should be used selectively). • Immediate feedback creates the perception of high performance. • The resulting feedback should be appropriate to the task.
Direct	Users must see the visible cause-and-effect relationship between the actions they take and the objects on the screen. This allows users to feel that they are in charge of the computer's activities.
Builds on Familiar Concepts	Each Web site should adopt elements of other common SFA Web sites as possible in order to leverage the user's knowledge across the entire range of SFA internet applications. There should be a sense of cohesiveness across SFA sites. Cohesiveness may be accomplished by reusing layout, headers, and navigational images.
Consistent	A consistent interface allows users to apply previously learned knowledge to new tasks. Effective applications are both consistent within themselves and consistent with one another. Within a Web site, fonts, colors, and images should be used consistently. Related applications and forms should work similarly.
Draws Eye Toward Important Areas	Avoid using bright colors or animation in a way that competes with content. Use color, fonts, images, and layout to focus attention on the key elements the page.
Avoids Redefining Commonly-used Behaviors of Elements	Doesn't redefine the behavior of elements the user knows from other contexts. For example, avoid using underline text for emphasis because users expect underlines in Web pages to indicate links.
Forgiving	Users make mistakes. User actions should be reversible. A good interface facilitates exploration and trial and error learning. For transactions, submit for an action with severe consequences should prompt users for a confirmation (e.g. for deleting user profile)
Breaks User Interactions into Small Chunks	Minimize the number of clicks necessary to access data or perform a function. For example, avoid organization structures within Web sites that are more than three levels deep. Map the site structure logically in ways that match the user needs and tasks.

2. Document Naming, Formats, and Specifications

2.1. Uniform Resource Locators (URLs)

A computer's name is composed of two parts, system name and the domain. The system name is the portion to the far left up to the first period (.), and the domain is the rest. So, the host *www.ed.gov* has a system name of www and a domain of *ed.gov*.

The left-most part of the domain name indicates the system name, which refers to a particular machine or resource.

- URLs should use simple, understandable words and be kept as short as possible without becoming cryptic. Use names that are easy to read, remember, type, and spell.

- URLs should be persistent/non-changing. Avoiding changing the URL of the web site.

- Ensure that the site's URL is visibly displayed on main web pages.

- Using lowercase for the web site URL.

2.1.1. Governing Authority

Since October of 1997, GSA has taken the sole responsibility of managing the Internet's .GOV and .FED.US zones. GSA registers second-level domains (between the last two dots) for government domains and requires an application process and justification for the creation of new domains. (See http://www.nic.gov/REFERENCE/). Their policy stalls the exhaustion of domain names in the .gov domain space and prevents conflicts between organizations that may both have a legitimate claim to a particular name.

GSA restricts the proliferation of domain names within a single agency. The most succinct form of the policy is "one agency, one name." (See “U.S. Government Internet Domain Names Policy” at http://www.nic.gov/REFERENCE/rfc2146.txt)

The following registrations will qualify for a separate .gov domain name:

- US governmental entities on the federal level only
- Agency listed in FIPS 95-1 with code ending in "00"
- Agency listed in US Government Manual
- Cross-Agency collaborative organizations
- Special Exceptions and requests are available

[NOTE: It may be possible to apply for the *sfa.gov* domain and have it approved by GSA, as SFA is considered to be a PBO and may be seen as autonomous enough for this type of request to be granted.]

There is no externally managed constraint, however to the number of **subdomains** that may be created between the resource and the second-level domain (See Section 2.2 – Subdomain Names).

2.2. Subdomain Names

[Provide SFA's strategy for subdomain naming schemes, use of aliases, etc. This section highly correlates to the sections on directory structure and directory naming.]

One method would be to develop SFA domain and server naming conventions patterned around the SFA organizational structure.

Using this strategy, SFA's primary domain would be identified by its acronym (SFA). [Need to determine whether to use "SFA," "OSFA," or "OSFAP."] For each office, provide an easy-to remember, yet easy-to-type, domain code (e.g. "COO" for the Office of the Chief Operating Officer, "CIO" for the Office of the Chief Information Officer, "STUDENTS" for the Students Channel, "SCHOOLS" for the Schools Channel, and "???" for the Financial Partners Channel. For example, the address of SFA's Students Channel page would be "www.students.sfa.gov," while the CIO's office address would be www.cio.sfa.gov.

[Need to develop a formal recommendation/strategy for SFA.]

- Subdomain names should use simple, understandable words and be kept as short as possible without becoming cryptic. Use names that are easy to read, remember, type, and spell.

- Subdomain names should help users visualize the site structure.

- Subdomain names should be persistent/non-changing.

- Subdomain names, as well as document and file names, should use lower case, to avoid problems with case-sensitivity. In general, people will not remember the difference between upper-case and lower-case characters. See also issue below (2.2) regarding case-sensitivity.

2.3. Directory Names

This section highly correlates to the sections on directory structure. May need to consider whether the two sections should be recombined. I split them apart because Directory Structure fit well under Site Structure, and Directory Naming fit well under Document Naming, Formats, and Specifications, but you'll also see it's difficult to talk about them separately.]

- To facilitate site management, and to make it easier for developers of one SFA web site to work on other SFA web sites, all SFA web sites should organize files into a standard directory structure (see Section 3.1 - Directory Structure) with a standard naming convention. Following is a table of standard directories, including naming convention and location on the server.

(The following examples were taken from the ED Internet standards document and used as a starting point for our own naming conventions – but we should come up with the SFA strategy. Also, if aliases will be used in our subdomain naming strategy, then we could have columns for directory name, alias name, location (directory structure), and description.]

Table 2-1: Standard Directories

Name	Description
/pubs/	A separate subdirectory for the HTML version of each publication.
/pdfdocs/	One or more separate files for each publication; subdirectories as needed to group related sets of PDF files.
/zipdocs/	A separate file for each zip file; subdirectories as needed to group related sets of zip files.
/offices/	A separate subdirectory for each principal office, each with its own subdirectories as needed; directory name is upper-case office acronym.
/news/	Archive of "What's New" announcements for past months.
/events/	Pages that announce events.
/images/	Contains shared graphics such as the SFA logo and ED seal, navigational icons, office logos, etc.; subdirectories for special collections such as state flags.
/comments/	Applications that solicit, collect, and display comments from outside customers.
/internal/	Web pages for use by SFA staff rather than outside customers. Could be a mirror of pages hosted on SFANet.

2.4. File Names

- Avoid arbitrary use of upper case (e.g., "picture.GIF" and "myFILE.html"). Standard SFA servers (e.g. Sun Solaris) are case-sensitive in its treatment of file names, and myfile.html is not the same as MyFile.html.

√ Never use spaces in a file name (e.g. "file name.html"). If two words in a file name must be separated, use an underline to join the two words together (e.g. "file_name.html").

√ Strict attention must be paid to matching actual file names with the links that refer to them, including ensuring that the correct (lower) case is used in the link.

[Should uppercase be allowed? See Section 13.6.1.]

2.5. File Extensions and Types

√ All static HTML files (Web pages that are not dynamically generated) in SFA Web sites must have an .html extension. All links to files on the server should reference .html extensions. DO NOT use .htm.

Δ Extensions are important and relate to the MIME types defined on the web server that delivers the files.

[Wayne's note: There might be a way to allow the use of both .html and .htm by including this information in the MIME header – the same may be true for .jpg vs .jpeg and .mpeg vs .mpg. Need to research/verify.]

[Sandy's note: I defined a standard html extension (.html) and excluded the use of .htm in this section to cut down on possible linking errors and to make document management more efficient. For example, when one developer links to another developer's page, if there is no standard extension type for html documents specified, the developer may erroneously mislink the document, and/or may have to check the document to which he/she is linking each time. This is quite time consuming. I believe the .jpg/.jpeg and .mpg/.mpeg is a separate issue and has more to do with the differences between platforms or products. We can discuss this further.]

√ The base part of the name should follow standard naming procedures for naming files (see Section 2.4 - File Names).

√ Authorized file extensions for use on an SFA Web site follow (see Table 2-1: Standard Directories).

[Need to link this section with authorized audio/video/plugins section (Section 10).]

Table 2-2: Standard File Types and Extensions

File Type	Extension	Example
Plain text	.txt	`<A HREF="servletsref.txt">Source Code for Java Servlets script</A>`
HTML document	.html (do not	`<A HREF="AboutSFA.html">About SFA</A>`

	use ".htm")	
Java Server Pages document	.jsp	`<A HREF="feedback.jsp">Feedback</A>`
Acrobat PDF	.pdf	`<A HREF="AboutSFA.pdf">About SFA</A> - PDF Version`
GIF image	.gif	`<IMG SRC="logo.gif" alt="Cool Logo">`
JPEG image	.jpg	`<IMG SRC="logo.jpg" alt="Cool Logo">`
WAV sound	.wav	`<A HREF="greeting.wav">Greeting from SFA</A>`
AU sound (μ-law files)	.au	`<A HREF="greeting.au">Greeting from SFA</A>`
Mp3 sound	.mp3	`<A HREF="greeting.mp3">Greeting from SFA</A>`
RealAudio sound	.rm	`<A HREF="greeting.rm">Greeting from SFA</A>`
AIFF sound	.aiff	`<A HREF="greeting.aiff">Greeting from SFA</A>`
QuickTime movie	.mov	`<A HREF="newSFAbuilding.mov">New SFA building</A> - QuickTime Movie (4MB)`
MPEG movie	.mpg	`<A HREF="newSFAbuilding.mpg">New SFA building</A> - MPEG Movie (4MB)`
WinZip file	.zip	`<A HREF="SFADocs.zip">Download all 3 documents</A> - WinZip file (1.2 MB)`

2.5.1. Documents Intended for Online Use

√ Material intended to be viewed, read, or browsed online must be prepared in HTML format (for text and tables) and GIF or JPEG (for images).

Δ GIF is recommended for graphics, line art and logos, while the JPEG format works best for photographic material or where there is a need to preserve a large number of gradient colors. In such cases, JPEG produces smaller files with minor loss of image precision. (see Section 9 – Graphic Design).

Δ Portable document formats such as Adobe Acrobat PDF should not be used as the primary format unless converting the material to HTML is not feasible. Although it is easier in many instances to create PDF documents rather than HTML pages, and while PDF preserves page layout and formatting, there are drawbacks:

- The contents of PDF files are not included in some full-text search indexes
- PDF viewers are not embedded in Web browsers prior to IE 4.0 and Netscape 4.0
- PDF viewers require more powerful hardware for online viewing than a Web browser alone

- PDF files present accessibility problems (Note: Adobe does offer web based services (see http://access.adobe.com) that enable sight impaired users to convert PDF files into formats, including HTML and plain text, that screen readers can understand and translate. Adobe, April 18, 2000 Press Release)

2.5.2. Documents Intended for Download or Offline Printing

Δ Offer links to printer-friendly and/or printer ready versions of long documents.

Δ The choice of file formats used should be based on the following considerations:

- The intended use of the material by the target audience
- The accessibility of the format to the target audience
- The level of effort required to convert the material to the format

Δ Material intended to be downloaded for offline print or display shall be prepared in one of the following formats, which are listed in descending order of preference:

1. HTML and GIF or JPEG
2. Adobe Acrobat (.PDF) (Must include link to downloadable free viewer.)
3. Microsoft Rich Text Format (.RTF) (RTF is easily created from most word processors and is more widely usable than native word processor formats such as Microsoft Word or WordPerfect. However, its reproduction of fonts and page layout can vary depending on the user's font set.)
4. Proprietary formats (e.g., WordPerfect, Microsoft Word, Excel, PowerPoint, etc.) should only be used if:
   A) Conversion to one of the above formats is not feasible
   B) The intended audience is known to have ready access to software which can handle the proprietary format
   C) The intended use is data analysis or manipulation (see below).

Δ If use of a proprietary format is unavoidable, use an earlier, more widely available version if possible.

2.5.3. Preparation and Use of PDF Documents

Δ PDF documents can be very useful for common forms and lengthy documents. If the intent is to print documents, PDF is often the preferred format.

√ PDF files should be accompanied by a link to the Adobe Acrobat PDF viewer download site. The following is the standard link (verify that it is current before using):

```
<A HREF="http://www.adobe.com/prodindex/acrobat/readstep.html">Adobe Acrobat
reader</A>
```

√ The document info tag must be completed. This becomes searchable text for a user coming to our site.

- Select: File-->Document Info-->General
- Complete all fields, including: Title, Subject, Author, and Keywords

√ General info is also where you will find information about file size and whether the file has been optimized. **The file must be optimized.** To optimize a PDF file, click on OK in the General Info box and then select File-->Save As. Before you save the file, make sure that there is a check mark in the Optimize box in the lower right hand corner. This will ensure that your file is optimized.

√ Bookmarks must be completed for all publications that are more than a couple of pages. In addition, when a customer opens the PDF document the bookmarks must be readily apparent. Here's how to make a PDF open with the bookmarks showing:

1. Open the PDF in Acrobat 4.
2. Select File, Document Info, and Open.
3. Change 'Initial View' from "Page Only" to "Bookmarks and Page".
4. Be sure to save the file via "Save As" with "Optimization" checked.

∆ If you do this to a PDF Version 1.2 file, it will become a 1.3 file. If you want to avoid the possibility of conversion glitches, you could use Acrobat Exchange 3 to change the Initial View setting for those files. The steps are very similar.

∆ If you have many files that have bookmarks, but that don't open with the bookmarks showing, you can use "File, Batch Process" to change just the Initial View settings and re-optimize the files.

∆ The addition of Thumbnails is especially useful for documents that have tables, charts or graphics. By looking at the thumbnails a user can easily identify where that information exists within the body of the document. By clicking on that page they will immediately go to it.

2.5.4. Documents intended for Download and Offline Analysis or Manipulation

∆ Material intended to be downloaded by the user for offline analysis or manipulation should be prepared in one of the following formats, which are listed in descending order of preference:

- Raw data files:
- Comma separated values, text in quotes

- Column formatted (mainframe data)
- Space delimited (purely numerical data)
- Proprietary formats (SAS, SPSS, etc.)

Δ Data files should be accompanied with adequate documentation of the file content and structure in a widely usable format such as ASCII or PDF. Documentation for executable programs shall include instructions for installation and use, as well as specifications for the platform needed to run the package (i.e., memory, disk space, operating system, etc.).

Δ If the material to be downloaded is large (i.e., larger than 500K), it should be compressed in ZIP format. ZIP files should be accompanied by a link to a standard directory containing zip utility software and instructions for unzipping the file on various computer platforms.

Δ A standard link for zip utility software offered on the U.S. Department of Education's Web site is:

```
<A HREF="gopher://gopher.ed.gov/11/info/utilities/">Zip Utility Software</A>
```

Δ Self-extracting archive files should not be used unless the contents of the package are only usable on one type of computing platform. [Is this still true?] Self-extracting program code only runs on one type of computing platform (e.g. Windows) and makes the contents of the self-extracting archive inaccessible to users of other types of machine (e.g., Mac, UNIX).

Δ Material in formats other than HTML shall be linked to an HTML page that describes the material in such a way that users of site-wide full-text search facilities can find material of interest.

2.6. File Sizes

Δ Generally, pages should be composed of no more than 100K, including images.

Δ Be aware that people may access your content over a 28.8 kbs modem, or less. Many users/customers will wait no longer than 10-15 seconds for a page to display. You should test your content over remote access lines to ensure it is accessible by those using dial-up services.

Δ One way to judge whether a file is too big is to test it using the connection method (14.4 or 28.8 modem, T-1 or T-3 line) that the majority of customers will be using. (Test results will also be affected by peak usage times and the capacity of the Internet Service Provider.)

Δ Estimated download times may also be obtained from some HTML editors (e.g. Macromedia Dreamweaver; Allaire HomeSite) or the Bobby validation service.

2.7. Information Priority

√ All Web documents must be evaluated for importance and priority. Information and links of high importance and priority should have greater prominence on the page. Communicate the priority of your content to your Content Approver. [Check whether information priority has been addressed in content management standards.]

2.8. Security Classification

√ All Web documents must be evaluated as Public, Internal, Confidential, or Privileged. Confidential and Privileged information must be clearly labeled. You may choose whether or not to label Public and Internal information.

[Verify that these levels and processes are consistent with SFA information security standards.]

√ Security access to a Web page should adhere to the most restrictive class of content on that page. For example, if a Web page contains both Public and Confidential information, then you should follow the Confidential usage and access restrictions.

Table 2-3: Security Classifications

Classification	Description	Usage Restrictions
Public	Information can be classified as Public only after it has been formally released for public distribution by an authorized SFA person or channel. Public information requires no protection against disclosure; it can be freely communicated with everyone.	None
Internal	Information must be classified as Internal if it does not clearly belong in any other classification (Public, Confidential, or Privileged). Most day-to-day information sources are for SFA's internal use only and should not be communicated outside SFA.	None
Confidential	Information must be classified as Confidential if it should not be disclosed to all employees; could damage SFA in some way if it were misinterpreted; or is considered private. [NEED TO FURTHER DEFINE THE MEANING OF "PRIVATE"]	
Privileged	Information must be classified as Privileged	• Publishing Privileged

	if SFA legal department or counsel created or directed the creation of information under attorney-client privilege. This classification has special protections under law that other classifications do not.	information requires the written consent of the legal department. • Web pages that send or receive Privileged information must be encrypted, the identity of the recipient confirmed by SFA Security Team, and an audit log maintained and archived for a period of no less than six months. • Privileged information and documents must be clearly labeled with the "Privileged" image.
For detailed descriptions of these categories, contact the Webmaster (e-mail: SFAWebmaster@ed.gov)		

3. Site Structure

3.1. Directory Structure

This section highly correlates to the sections on directory naming. May need to consider whether the two sections should be recombined. I split them apart because Directory Structure fit well under Site Structure, and Directory Naming fit well under Document Naming, Formats, and Specifications, but you'll also see it's difficult to talk about them separately. Also, there are some crossovers with the Subdomain naming schemes section.]

√ The index.html standard will be used wherever possible as the default entry point for a directory.

√ URLs should be "hackable" to allow users to move to higher levels of the information architecture by hacking off the end of the URL.

√ Follow directory naming conventions outlined in Section 2.3 – Directory Names.

[You could develop this section further by creating a subsection on each major type of folder category, like the following Images section.]

3.1.1. Images

√ All top-level images [Need to define "top-level image."] are located in the common "images" folder of the root directory, and the sub level images are in their respective images folder. Please refer to the Section 2.3 – Directory Names for more details.

√ All common graphics in a Web site must be put in a common images directory (e.g. "/images") and should be made available for all SFA developers to use. Please refer to the Section 2.3 – Directory Names for more details.

Δ You must use these images as provided, without altering the dimensions of the logo by using different WIDTH and HEIGHT attributes, and without modifying the image. [What process should a developer use to admit a modified or new image to the common images folder? Is this covered in content management standards?]

4. Navigation

4.1. Consistent and Predictable Navigation

√ The style of navigational buttons should be consistent throughout sections of a Web site or application.

√ The fonts used for text and on navigational graphics and graphical buttons should be consistent throughout the site and should be easy to read. Arial is recommended as a legible font that will work across browsers.

√ Navigation bars should be formatted consistently.

√ Navigation bars should be consistently placed throughout the site. Ideally, a horizontal global navigation should be placed at the top of the page.

√ Ensure that all scripts do not disable the "Back" button on the browser.

4.2. Efficient Navigation

√ Navigation bar should indicate user's location within the site (e.g., navigation bar links change color relative to user's current position in the site, 'bread crumb' navigation list).

√ To assist users in navigating "sectioned" documents, each page should include standard navigation buttons at the bottom with links to the table of contents and previous and next section. The first and last sections of the document should omit the previous and next link respectively.

4.3. Frames

√ Frames are not to be used on SFA Web sites. Frames interfere with accessibility for persons with disabilities. Instead of using frames to organize content, use tables or INCLUDE files. See [NAME OF SOURCE DOCUMENT] for the current list of products and standards, including a list of approved HTML editors, for SFA Web application design and development.

4.4. Image Maps

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Navigation – Image Maps."]

Avoiding large image maps
Considering the user when designing image maps
Providing hotspot text for image maps
Providing mouse-over text for image maps
Providing titles for image maps
Supplementing image maps with text links
Using client-side vs. server-side image maps
Using image maps for efficient navigation
Using image maps for small elements
Using image maps to change the page appearance

4.5. Links

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Navigation – Links." However, you'll need to note and resolve the overlap with the "Links" section.]

Avoiding broken links
Avoiding embedded links on web pages
Avoiding linking from every occurrence of a word
Avoiding links within the content
Describing the destination of the link
Getting permission to use external links
Grouping links into short lists
Keeping linked text short and concise
Keeping the link style consistent
Limiting the number of links to get information
Matching the link text with the destination
Placing external links deep in the web site
Placing links consistently
Providing a link to the webmaster
Providing links to basic site areas
Providing links to higher levels
Showing images as links
Using links to navigate through long web pages
Using non-standard link colors
Using only relevant external links

Using Top of the Page links

4.6. Mouse Users

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Navigation – Mouse users."]

Avoiding double click only operations
Avoiding mouse plus keyboard shortcuts
Avoiding moving targets
Providing hot zones around small objects

4.7. Navigation Buttons

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Navigation – Navigation Buttons."]

Avoiding broken navigation button links
Avoiding Next and Previous navigation buttons
Keeping navigation buttons consistent
Making navigation buttons fit the web site theme
Using navigation button bars

4.8. Navigation Menus

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Navigation – Navigation Menus."]

Balancing the navigation menu structure
Designing local navigation menus on web pages
Extending global navigation to sub-sites
Using cascading navigation menus for large sites

√ Global navigation bar should include a link to the home page.

√ Global navigation bar should include a link to primary site support tools (site map, site index and/or site search tool).

4.9. Search Utilities

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Navigation – Search Utilities."]

Making the search utility search the whole site
Providing a search utility
Providing alternatives to finding information
Using a search utility for navigation

4.10. Site Maps

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Navigation – Site Maps."]

Making the search utility search the whole site
Providing a search utility
Providing alternatives to finding information
Using a search utility for navigation

5. Links

5.1. Use of Links in Page Content

- Ensure that link names effectively and accurately describe the target page. Do not use the phrase, "click here," or any other version of a link phrase.

- Ensure that content links are grouped logically.

- Ensure that duplicated link names connect to the same content.

- Beware of placing excessive links in your paragraphs of text. Too many links will make the page unreadable and hard to scan. Less is more. If you have several links in a paragraph, consider turning them into a bulleted list of links.

- Place link anchors on the most relevant word(s) in the sentence; do not underline the entire sentence.

5.2. Absolute and Relative References

An **Absolute Reference** provides the complete http address in the form of a full URL, including the server name and complete path. For example, a file named "map.html" in the `docs` folder within the `project_1` folder would be linked as an absolute reference as follows:

```
<A HREF="http://www.ed.gov/sfa/project_1/docs/map.html">SFA Map</A>
```

Absolute references only work if the basic structure of a web site and its www directories do not change. If the web team were to move a project's Web pages to another directory, the reference would no longer point to the correct location.

An **Absolute Reference from the Root Directory** is a pathname that starts from the root directory, followed by the pathname required to access the file or directory from the root directory. An absolute link begins with a slash (e.g. "/") and is called an absolute path since you are giving the explicit path to the object from the root. A **root directory** is the "/" directory that is at the root of the server's filesystem tree.

For example, an absolute path to a page containing a map located in the docs directory on a sample development site would be linked from the root directory ("/") through the path to the docs folder:

```
<A HREF=”/sfa/project_1/docs/map.html”>SFA Map</A>
```

A **Relative Reference** can be defined as one that connects to a location relative to its current location. The link provides a partial http address in the form of a fragment of a partial or full directory pathname and file name.

```
<A HREF="map.html">
```

Creates a link to a file by the name of `map.html`. This link, though simple in construction, is a basic relative link. It will link to a page in the same directory as the current page. By default, if no server name or path is specified in html, the file reference is to the default current directory, which in most cases is the folder housing the document containing the link.

```
<A HREF="docs/map.html">
```

Creates a link to `map.html` within the `docs` directory. This link is a relative link as it refers to a directory and file relative to the current page.

```
<A HREF="../map.html">
```

Creates a link to `map.html` one directory above the current directory. This link is a relative link as it refers to a directory and file relative to the current page. To move from a lower-level folder (i.e. a folder within a folder) up in the direction of the root folder, use `"../"` for each folder you want to climb.

The advantage in using relative references is that if you move a set of Web pages to another directory, the references to other files that were moved are still valid without having to change every reference. For example, a developer might author twelve related SFA documents in a local directory of a computer (development environment) and then move all of them to the Web server. The developer might not even know the name of the server subdirectory on which they will reside.

The eventual location of the documents does not matter when using relative links. The developer would complete the documents, link them to each other, and test without knowing what the absolute URLs will be. In this case, if the developer used full URLs, he/she would have to know the full path on the server where the documents would be housed. If they were moved, someone would have to edit each document within the set.

5.3. Internal Links

- All internal links should be coded as a relative reference, rather than an absolute reference or absolute reference from the root directory.

√ Links that are local within the same server directory should be added as follows, using relative URLs:

```
<A HREF="filename.html">link description</A>
```

The use of relative links will allow a developer to test locally on a machine and will not require any changes when the files are uploaded to the Web server.

5.4. External Links

[Wayne's note: Need to research and adopt the use of the XLink standard (Relative External References) for linking to external links.]

√ Only references to external Web links or links to sites residing on other servers should employ the use of absolute URLs. For example, a link to OPM's Federal Leave Policies site might be:

```
<A HREF="http://www.opm.gov/oca/leave/index.htm">Federal Leave
Policies(OPM)</A>
```

√ Test all links to ensure that the URL is current and that the information loads in a reasonable amount of time.

Δ Do not include links to sites with unusually long load times (see Section 5.5 – Links to Large File Sizes).

5.5. Links to Large File Sizes

√ Links to files larger than 30 kilobytes, such as images or PDF files, must be labeled with the file size to inform users of a potential delay in downloading, e.g. SFA Org Chart (35K).

5.6. Links to Images and Image Maps

√ When linking from images and image maps, provide alternative navigation techniques such using <ALT> tags for images and mirroring image maps with text links.

5.7. Keeping Current on Links

[Wayne's Note: Need to create and document a regular maintenance process that validates all external links – check into TeamSite functionality.]

- When linking to existing content on other pages, contact the owner of the page to which you will be linking as a courtesy. Be aware that people may change their content, so you should check those links periodically.

- When linking to a document that was created dynamically (e.g., with a database), be aware that those page names are automatically generated, which means that your link may point to the wrong place if the document is edited and reconverted.

6. Standard Broad Functional Categories

Functional pages described in this section, including utilities such as the site map and search page, are consistently found in Web sites and should be included in all SFA web sites where appropriate.

Pages that fall under this category are usually included in the Web site's main functional navigation bar, which is located at the top of the Web page in the menu of broad functional categories.

6.1. Site Maps and Table of Contents

√ Each Web site or Web application should maintain a Site Map that shows the structure of the site. Site maps help both the developer and end user understand how information is organized within a site.

√ The Site Map should list the main content areas of the site and should hyperlink from the Site Map to these areas. An extensive listing or graphic overview of internal site links offers users quick access to your content.

Δ Online, a site map is equivalent to a table of contents found in a print document, while the site search tool functions as an index. Search facilities are not a substitute for a clear, well-organized list of a site's contents, but search facilities should be used as a supplement to the site map. Resources are better spent on a robust site search tool than maintaining an extensive site index.

Δ Text-based site maps, in the style of a table of contents, are generally more efficient and informative than a graphical metaphor, such as a hierarchical branching diagram. Furthermore, graphical site maps are more difficult to maintain, such as when a site is reorganized or new information is added or deleted.

√ Modifications to the Site Map should be coordinated through the primary Web manager or Web developer.

6.2. "What's New" Pages

In the first few years of the World Wide Web, "what's new" pages were used extensively to indicate in a central places all changes to content that had been made during a certain time frame.

Δ Most Web sites are now continually updated and refreshed to keep content "fresh." In order to inform users about new content, you could place a "new" graphic next to each updated item.

Δ Sites that are complex, with many levels of information spread over a great number of pages may prefer to maintain a "what's new" page, rather than posting "new" graphics everywhere.

6.3. Search Features

Keyword searches allow the user to search for specific information that they know exists but are not sure where it may be located in the hierarchy of the Web site, or if the information they are looking for is not listed in the site's table of contents or site map.

√ All SFA sites should offer a robust, within-site search facility for users. The current standard search engine for SFA is Autonomy's Knowledge Server (see SOURCE DOC).

√ Indicate whether the search engine searches the entire Web site, a collection of documents (such as a subsection), or a database. Sites that are particularly complex should offer readers a menu that allows them to limit their search to a specific area.

√ The site search tool should provide advanced search options or features (e.g., match all/any words, exact phrase searching – can use quotes, Boolean searching – can use AND/OR/NOT operators, etc.).

√ The site search tool should provide the ability to narrow a set of returned results (e.g., perform a sub-search, search within, etc.).

√ The site search tool should provide the option to sort search results (by relevance, date, document type or other criteria).

√ Provide a help page with search tips and/or instructions on using the site search tool.

6.4. Contact Information and User Feedback

√ Provide links to contact information, in a prominent contact page in each site. The following information is typically provided on a contact page.

- Important Email addresses
- Street addresses
- Technical support phone numbers
- Organizational phone numbers
- Fax numbers
- Maps, travel directions, or parking information

- The SFA Webmaster email address, used for technical Web site support, should be posted in the footer of every Web page (see section on headers and footers).

- Provide a Web form to collect user information and feedback. A database may be used to store and analyze user input. Be certain to have a Web site customer service infrastructure in place to handle incoming questions from users.

6.5. FAQ Pages

Frequently asked questions pages offer a central location for the most commonly asked questions received from users.

FAQ pages are primarily used to provide self-service support and information and are intended to improve users' understanding of the information and services offered while reducing the burden on SFA support staff.

7. Web Content

7.1. Editorial Style

7.1.1. Keep Content Concise

- Online documents should be concise and structured to make it easy for Web users to scan the page quickly. Most customers are looking for information, rather than reading a Web page word for word.

- Keep headings, lists, and other elements that are intended to grab a user's attention clear and precise.

- Place important facts and conclusions at the beginning of a text where users and the search engine can find them quickly.

- Avoid the use of extraneous content.

- Avoid displaying long text passages.

- Provide titles for chunks of information.

- Capitalize all headings and subheadings.

7.1.2. Use Appropriate Language for the Audience

- Use language that is familiar to the target audience.

- Avoid the use of nonstandard language.

- Avoid the use of jargon and slang.

- Identify and expand acronyms and abbreviations.

- Avoid the use of passive voice.

- Make the language internationally compatible.

- Make dates internationally compatible.

- Use Universal Time Coordinated format.
- Use appropriate units of measurement.
- Use consistent terminology.

7.1.3. Metaphors

- Choose an interface metaphor appropriate for the audience.
- Avoid using multiple or conflicting metaphors.
- Avoid using too many metaphors.

7.2. Copyrights

7.2.1. Publishing in the Public Domain

[Explain further. This section explains that all government-sponsored material is in the public domain and cannot be copyrighted.]

7.2.2. Respect Copyright Law

- Ask Permission to Use Logos and Trademarks. [Explain further.]
- Using a Copyright Page. [Explain further.]

7.3. Misspelled Terms and Phrases

- Avoid misspelling the word, "Internet." [Explain further.]
- Avoid misspelling the word, "Web." [Explain further.]

7.4. Misused Terms and Phrases

- Avoid misusing the term, "return to." [Explain further.]
- Avoid misusing the term, "point your browser." [Explain further.]

- Use confusing or misusing the terms, "WWW" and "Internet." [Explain further.]
- Avoid misusing using the term, "cancel." [Explain further.]
- Avoid using the term, "previous." [Explain further.]
- Avoid using the term, "surfing." [Explain further.]
- Use confusing or misusing the terms, "site" and "sight." [Explain further.]
- Use confusing or misusing the terms, "back" and "forward." [Explain further.]
- Use confusing or misusing the terms, "enhanced" and "corrected." [Explain further.]
- Use confusing or misusing the terms, "view" and "read." [Explain further.]

7.5. Overused Terms and Phrases

- Avoid overusing the phrase, "and more…"
- Avoid overusing the phrases, "hot links" and "hottest of cool."
- Avoid overusing the terms, "cool," "hot," and "neat."

7.6. Unnecessary Terms and Phrases

- Avoid unnecessary link terms and phrases. [Explain further.]
- Avoid the unnecessary phrase, "come back often." [Explain further.]
- Avoid the unnecessary phrase, "cutting edge." [Explain further.]
- Avoid the unnecessary phrase, "feel free." [Explain further.]
- Avoid the unnecessary phrase, "click or press this button." [Explain further.]
- Avoid the unnecessary term, "click." [Explain further.]
- Avoid the unnecessary term, "current." [Explain further.]
- Avoid the unnecessary term, "note." [Explain further.]

- Avoid the unnecessary terms, "describes" and "here is." [Explain further.]

7.7. Content Maintenance

- Avoid the occurrence of dead end web pages. [Explain further.]

- Avoid the use of "Under Construction" pages. [Explain further.]

7.8. Spell Checking/Proofreading

- All documents, as well as text on images, must be carefully and thoroughly checked for spelling or grammatical errors before posting them. Refer to the Associated Press (AP) Style Guide for spelling and grammar.

[Wayne's Note: NEED TO CREATE A COMMON "SPELLCHECK" DICTIONARY FOR SFA AND SUBJECT AREAS – Note: most HTML generators allow you to "plug-in" a common spell-check function (e.g. MS Word) which in turn allows the developer to specify a common dictionary. SFA should take steps toward building that common dictionary.]

8. Page Structure and Appearance

Throughout your site, you should have consistency not only in your layout grid, but how you present your images, the style of your navigation bar, the fonts and functionality of your site. A consistent layout makes your site more intuitive, visitors more comfortable and more likely to continue navigating your site. A consistent interface and branding lets them know that they're still at your site.

However, you should still make an effort to provide differences between sections or subsites so the visitor can distinguish distinct areas in your site. These variations between sections/subsites aid in navigation as it helps the visitor to know where they are in your site.

8.1. Functionality vs. Web Page Design

- Considering the use of the page in the design [Explain further.]
- Meeting objectives vs. web page design [Explain further.]
- Placing web page content to catch attention [Explain further.]
- Consider the target audience [Explain further.]
- Design web pages for the lowest common user [Explain further.]
- Provide quality content on all web pages [Explain further.]
- Give good information on the first two web pages [Explain further.]
- Use a consistent design theme on your web pages [Explain further.]
- Avoid cliches on your web pages [Explain further.]
- Avoid novelties on web pages [Explain further.]
- Avoid overdoing a web page [Explain further.]
- Keep web pages fresh [Explain further.]
- Provide print-only versions of web pages [Explain further.]
- Planning web page redesigns [Explain further.]

8.2. Alignment

Δ Text in the main body of a web page should be left aligned. Centering paragraphs of text on a page makes the content hard to read.

√ Columns of numbers must be right justified. Generally this is easiest to do within TABLES. Numbers within a column must have the same number of decimal digits so they align on their decimal points.

Δ Avoid using tables that lay out text in parallel, word-wrapped columns.

8.3. Font Styles

Δ Use typographic styles sparingly to ensure a clean, consistent flow to the content. No more than two fonts should be used on one page.

Δ Font styles should be used consistently throughout your site.

Δ There should be sufficient contrast between one font and another, and between text blocks, headlines, and the surrounding white space.

Δ Font style enhances readability. Generally, it is recommended to use **san serif** fonts (e.g. Times New Roman) as the body text font. Use **serif** fonts (e.g. Arial, Helvetica) for headlines and accent text.

Δ Times Roman, Arial, and Helvetica are the default fonts for Netscape and Internet Explorer web browsers and should be used in the design of all web pages to ensure consistency in appearance.

√ For each font, specify alternative fonts, in case the font is not installed in the user's collection. Every list of fonts for the browser to choose from should end by indicating a generic font style, either "Serif" or "Sans-Serif."

Δ Avoid monospaced fonts.

The following table identifies fonts included in default collections that are installed with Macintosh or Microsoft Windows and from the Truetype fonts that Microsoft has made publicly available online at http://www.microsoft.com/opentype/.

Table 8-1: Default Font Collections

Core Fonts for the Web	Installed as Default Font in the Following Products:							
	Win95	Win98	Win98 SE	Mac 7.0	UNIX	Netscape 4.0	IE 4.0	IE 5.0

Arial	X	X	X			?	?	?
Comic Sans MS	X	X	X			?	X	X
Courier New	X	X	X	X	X	X		
Georgia							X	X
Helvetica				X	X	X		
Symbol	X	X		X	X			
Times New Roman	X	X	X	X	X	X		
Trebuchet							X	X
Verdana		X	X				X	X
Webdings	X	X	X			?	X	X

8.4. Font Size

- Larger Font sizes enhance readability. Make sure the font size is large enough to be legible for all audiences.

- Explain further how to determine which point size of text to use.

- Specify the minimum point size for screen fonts.

8.5. Font Color

- Darker font colors enhance the readability of text. Ensure that font and background colors are high contrast. The ideal combination is black text against a white background (which also ensures that the user will be able to read the text if he/she prints the page).

- Pay particular attention to link colors against a non-white background. It is generally recommended to set link colors to default (link= “blue”, active link=”red”, visited link=”purple”).

- Avoid using color to emphasize text.

8.6. Heading Tags

- Use headings sparingly to ensure a clean, consistent flow to the content.

- There should be only one <H1> heading per page. The page name specified by the <H1> tag generally should be identical to the title specified in the <TITLE> tag. The H1 tag represents the main theme/idea/purpose of the page.

- A graphic with ALT text may act as a header. Other HTML headers used on the same page should take into account the relative size of the graphic and be sized accordingly. For example, a

graphic banner might replace the top-level <H1> on a page; therefore, any HTML headers used below it would be <H2> and smaller.

- Lower-level headers (e.g., <H2>, <H3>, etc.) may be used if appropriate to the document.

- Documents which are divided into multiple pages should include the document title and publication date in italics above the top-level header, to help identify the document to users who may arrive at the page without knowing its context, (e.g., the result of a full-text search).

- Use <H4>, <H5>, and <H6> for captions, or when you have many levels of sub-headings. In general it's best not to go lower than <H4>.

- Header markup should not be used to emphasize entire paragraphs. Generally, section titles and other text marked with the Subhead style in a word-processing document should be marked up as lower-level HTML headers instead of simply appearing in bold.

- Cascading Style Sheets may be used for specifying the appearance of header tags.

- Additional text formatting tags (bold, italic, etc.) should not be applied to headers because they can confuse some browsers (like Lynx). No validity (syntax) checker will permit inappropriate use of heading tags because they are intended to convey the outline or structure of a document. Most browsers default to bold to display headers anyway.

8.7. Text Formatting

- When creating a new document using a word processor that will be converted to ASCII or HTML, do not use the word processing program's style sheets to produce "all capitals" or other formatting effects. The effects will be lost during the conversion.

- No text should be underlined unless it is a link. This includes text that appears on graphics as well as typed text and headings. Be particularly aware of the need to make modifications to the use of underlined words and sentences when converting existing word processing documents to HTML.

- Animated, blinking or zooming effects should not be applied to text.

- Avoid using all uppercase text.

- Avoid the use of italicized text.

- Specify the policy on creating artistic or special fonts.

- What is the policy on varying text for effect?

8.8. Tables and Grids

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Application Design – Tables and Grids."]

Aligning table column headings
Avoiding black borders around all table cells
Avoiding horizontal scroll bars in grids
Controlling the line length using tables
Defining the cell widths of tables
Height of column headings in tables
Laying out table information
Making every fifth table row blank
Ordering items in tables
Organizing web pages using tables
Placing web graphics using tables
Providing column and row headings for tables
Providing keyboard navigation of grids
Using blank table rows/columns for white space
Using grids to correlate variables

8.9. Page Width

√ Use relative table widths and percentages when designing the width of a page. Do not design pages that are dependent on a fixed screen width or use absolute pixels. Screen width depends on the user's monitor size, browser and operating system.

√ As a widely used government agency, SFA Web pages must meet the needs of as many users as possible. With that in mind, all SFA Web pages must be created to accommodate users with low screen resolutions. This minimum level of screen resolution for Web applications is a resolution of **800 pixels wide by 600 pixels high** – the VGA standard resolution. Even though most graphic cards and monitors now support greater resolution, many users leave their machines configured at VGA resolution because of small (less than 17") screen size. It is easy to forget this in a Web development environment with more advanced machinery.

Δ At a minimum, ensure that your design degrades nicely to work with a minimum page width of around 470 pixels.

8.10. Page Length

√ The home page of a Web site should be less than 1.5 screens in length (also see Section 11.1 - Home Page).

√ For other web pages, two to three scrollable regions are generally acceptable (roughly equivalent to one printed page). Large or complex documents *intended for online viewing,* typically those larger than two screens in length, should be divided into multiple, smaller files with a menu page or Table of Contents describing the various sections. Page length should be based on the standard screen resolution for the SFA client platform.

∆ Web pages longer than two screens that are intended for both online viewing *and* printing should contain navigational links to eliminate the need for scrolling to navigate. A table of contents should be provided, along with <A NAME> anchor links.

∆ If possible, files should be divided along logical break points such as chapters or sections.

8.11. White Space

∆ Although the horizontal rule (<HR>) is a popular tool of HTML and makes it easy to create divisions between sections, avoid using horizontal rules if at all possible. The use of horizontal rules can make the screen look cluttered and detract from the overall design and readability of the page.

∆ In place of horizontal rules, use 'white space' to create divisions between sections on a page. White space is a blank area between sections and is may be achieved by inserting a clear single pixel and giving it the desired height or width dimensions.

∆ Use <BR> when you need to break a line but leave no blank space. The <BR> tag is used similarly to a RETURN key in a word processing program.

∆ To skip a line, use the paragraph tag, <P>. However, a sequence of multiple <P> tags is treated as a single blank line. If you must insert multiple blank lines, use the <PRE> and </PRE> tags with blank lines inserted between them.

9. Graphic Design

9.1. Backgrounds

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Graphic Design – Backgrounds."]

Avoiding high-intensity colors as backgrounds
Avoiding photographs as backgrounds
Avoiding watermarked text backgrounds
Choosing patterned backgrounds
Choosing solid color backgrounds
Keeping patterned backgrounds light in color
Keeping the background image file size small
Using complex or patterned backgrounds
Using random images for patterned backgrounds
Using solid background colors
Using white backgrounds to improve legibility

9.2. Colors

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Graphic Design – Colors."]

Avoiding blue text
Avoiding relying on color to give visual clues
Avoiding similar text and background colors
Avoiding using only color to identify an item
Using bold colors for emphasis
Using contrasting text and background colors
Using neutral colors for minor elements
Using no more than 4 non-neutral colors at once
Using primary colors for charts and graphs

√ All pages must have a light colored background and dark colored text. The navigational framework within sites may be the one area that departs from this standard. Regardless, there should be adequate background/foreground contrast to ensure legibility.

- Background and text colors must remain constant throughout any given Web site.

- More than two or three colors of highlight will generally lead to a confusing interface. If that many colors are required to make your message known, you should consider giving the user the capability to limit the information they have to deal with at any given time.

- For the sake of those who might be colorblind, choose colors not only on the basis of base-color (Red vs. Green), but also on basis of contrasting brightness. Consider whether another interface device (use of an icon, for instance) could convey the same message. For more information, see SFA Accessibility Guidelines. Also be aware that some color combinations do not work well together. Avoid saturated blue/red (text to background combos) – as they cause chromatic aberration and are difficult to read

9.2.1. Color Palettes

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Graphic Design – Color Palettes."]

Showing relations using color
Using a consistent color palette

- All non-JPEG graphics and color elements should use the 216-color cross-platform Web palette. Use the palette to create all blocks of colors, text, lines, etc., and to reduce the color depth of images. Using the standard palette prevents images from dithering, or looking dotted and fuzzy, in the browser.

9.3. Graphic Formats

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Graphic Design – Graphic Formats."]

Avoiding problems anti-aliasing transparent GIFs
Avoiding proprietary graphic formats
Considering quality vs. compression of JPEGs
Converting full-color photographs to GIFs
Ensuring the text quality of JPEGs
Managing changes to graphics
Optimizing the color palette of GIFs
Selecting a transparent color for transparent GIFs
Using compression with JPEGS to reduce load time
Using custom color look-up tables with GIFs

Using GIF format for images with text
Using GIF format for simple images
Using interlaced GIFs
Using JPEG format for large images
Using JPEG format to preserve true color
Using progressive JPEGs
Using the system palette for GIFs
Using transparent GIFs

√ Images should be saved in either the Graphic Interlaced Format (GIF), including GIF, GIF89, and animated GIFs, or Joint Photographic Experts Group (JPEG) format.

√ Name images according to the SFA "File Extensions and Types" and "Document Naming, Formats, and Specifications" standards.

9.3.1. GIF vs. JPEG Format

Δ GIFs should be used for images that predominantly consist of solid colors, blocks of colors, lines, and/or text. The Graphics Interlaced Format reads images line by line from left to right. Therefore, the more color variations per line, the greater the size of the GIF file. GIFs use no more than 256 colors, of which only 216 are Web-safe; try to use even fewer if at all possible (especially for buttons or simple graphics or icons).

Δ If an image has many color variations, and if attempts to "GIF" this file using the 216-color Web palette results in substantial quality loss, use the JPEG format. The JPEG format is especially efficient for images made up of many colors, complex gradients, and photographs. All photographs (except small duotones or grayscales) should be compressed using the JPEG format. If you are constrained for space, JPEG your image at the "Medium" setting. If you require better quality at the price of a higher file size, JPEG at the "High" setting.

√ GIF images must be saved as "Interlaced" instead of "Non-Interlaced." This makes the page appear to load faster for users with slower connection speeds. The image will be displayed roughly at first (via intermediate pixel lines) and will become more intelligible as the browser receives more data.

9.3.2. GIF File Size

Δ Reduce the number of colors used in a GIF graphic to the greatest possible degree without affecting quality of the image. Reducing the number of colors in the GIF reduces the size of the file and consequent download time.

To reduce the bit size (number of colors used) of GIF images:

- To reduce the size, "index" the image to the fewest numbers of intermediate colors possible. Start low and work up the number of colors that looks best.

When creating a GIF graphic for the Web, keep the file size smaller by using non-dithering (non-speckled) settings. A non-dithered graphic reduces the number of colors in the GIF, which in turn, reduces the size of the file and consequent download time. Non-dithered graphics also tend to appear cleaner and less amateurish.

To create non-dithered GIF images (using Adobe Photoshop 5.5):

1. Create your image.
2. Select File > Save for Web…
3. The Adobe ImageReady window will appear. In the "Settings" drop down menu, select "GIF 128 No Dither."
4. Check the "Interlaced" box.
5. Adjust the color depth if you need more or less colors in your graphic.
6. Click "OK" to Save.

9.4. Orders around small icons

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Graphic Design – Orders around small icons."]

Avoiding culture-specific icons
Avoiding cute or clever icons
Avoiding hands or fingers in icons
Choosing colors for icons
Considering the use of the icon in the design
Designing icons that are simple and clean
Indicating hot icons
Sizing icons
Sizing toolbar icons
Using familiar symbols for icons
Using icons consistently
Using perspective in icons
Using the standard 16-color palette for icons

9.5. Images

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Graphic Design – Images."]

Avoiding bare body parts in images
Avoiding humorous images
Avoiding images from overused libraries
Avoiding overusing images
Avoiding racial characteristics in images
Avoiding relations between sexes in images
Avoiding using only images
Ensuring the quality of the images
Keeping images consistent
Making separator bars 3 pixels in height
Reusing graphics to increase page load time
Supplementing images with text
Using a web-safe color palette for web images
Using borders to distinguish images
Using images for titles
Using images to enhance the application
Using minimal detail in images that are drawings
Using splash screen images
Using thumbnail images

- Use images sparingly and only if they add value to the page or accompanying text.

- Images should not block any important information.

- Image should load last and not impede the loading of the page.

- If using a background image, provide a similar background color as the background image loads.

9.6. Image Attributes

- Provide alternate text for graphics. All images must contain <ALT> text in case the image does not load in the browser, or in case it loads slowly. Including <ALT> text also ensures accessibility for users with text-only browsers, browsers with images turned off, and screen-readers.

- Images must contain the WIDTH and HEIGHT attributes to help images load faster.

9.7. Image Maps

- Image maps should have clearly defined “clickable” areas so users are not confused about what will happen when they click on an image map.

- √ Provide text links for all active regions of an image map.

- Δ Use client-side image maps rather than server-side image maps.

9.8. Layout Principles

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Graphic Design – Layout Principles."]

Avoiding text-only layouts
Choosing the length of the layout
Considering the proximity of items in the layout
Creating visual impact in layouts
Designing layouts for small monitors
Indicating importance in layouts
Keeping layouts conservative and simple
Keeping section titles less than 7 words
Keeping text lines to less than 10 words
Limiting the amount of text in layouts
Making the layout aesthetically pleasing
Placing text in chunks in layouts
Using grids for layouts
Using hanging indents in layouts
Using horizontal space in layouts
Using items that complement each other
Using separator bars in layouts

10. Multimedia Usage

10.1. Animation

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Multimedia – Animation."]

Avoiding non-GIF animation
Avoiding replaying animation
Choosing the right amount of animation
Considering the user when using animation
Displaying animation in a separate window
Keeping animation images small
Making animation beneficial
Making animation relevant
Using a slide show animation
Using animation to draw attention
Warning about animation

√ Provide text equivalents for animations. For more information, see SFA Accessibility Guidelines.

10.2. Audio

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Multimedia – Audio."]

Avoiding replaying audio
Avoiding symbolic music
Choosing the right amount of audio
Considering the user when using audio
Making audio relevant
Making sounds functional
Providing an option to turn off the audio
Synchronizing the audio and video
Using audio for web site content
Using audio to evoke a mood
Using only high-quality audio
Using realistic sounds

Warning about audio

√ Use cross-platform file formats, such as Mp3, RM, and AU for non-streaming audio.

Δ WAV and other platform-specific formats may be used if alternative formats of the same file are provided for users of other platforms.

√ All audio components must include ALT tags for text-only browsers and screen-readers. For more information, see SFA Accessibility Guidelines.

√ You should provide alternatives to audio or video content, such as a text-based transcript. For more information, see SFA Accessibility Guidelines.

10.3. Plug-Ins

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Multimedia – Plugins."]

Avoiding plug-ins
Placing plug-ins on a separate page
Providing instruction on plug-ins
Warning about plug-ins

√ The following formats requiring plugins are acceptable for use on SFA web sites:

- Macromedia Flash (.swf)
- REAL Video and Audio (.rm)
- Acrobat Reader for PDFs (.pdf) (see info on PDF Accessibility issues, downloads, etc.)

√ Notify users of any necessary applications or hardware required to view information on the site (e.g., You need Flash to view content on this site, Best viewed with v.x of Netscape, etc.). Provide a URL and a link for the user to download the appropriate plugin.

√ You must provide alternatives to files requiring plugins, such as a text-based transcript. For more information, see SFA Accessibility Guidelines.

10.4. Video

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Multimedia – Video."]

Avoiding video on Internet sites
Choosing the right amount of video
Considering the user when using video
Making video relevant
Synchronizing video and audio
Warning about video

- Use cross-platform file formats, such as MOV or MPEG for non-streaming video.

- All video components must include ALT tags for text-only browsers and screen-readers. For more information, see SFA Accessibility Guidelines.

- You should provide alternatives to audio or video content, such as a text-based transcript. For more information, see SFA Accessibility Guidelines.

10.5. Applets and Programmable Objects

- Provide text equivalents for applets and programmable objects. For more information, see SFA Accessibility Guidelines.

11. Application Design

Consistent and easy-to-navigate layout is essential to good Web page design. Use the sections below as guidelines for developing clean, simple, and accessible page design.

Web site design templates are provided for you to use as a general guideline when you create your Web pages. You should use the general layout approach detailed in this section to arrange your content on a Web page. These templates contain information about standard areas, headers, and footers that should appear on every SFA Web page.

11.1. Home Page Design

The home page is the first page of a Web site that users see. The home page is often a launching pad for the other sections of the Web site.

- Make the home page relevant to the site and audience.

- Include major audience and purpose links on the home and make sure they are organized logically on the home page to facilitate quick access to desired information.

- Provide a link to “new site user” information on the home page.

- Place the global navigation bar on the home page.

- Provide a link to the site map on the home page.

- Avoid external links on the home page.

- Provide option to search or a link to the site search tool page on the home page (See Section 6.3 - Search Features).

- Provide a link to agency profile on the home page.

- Provide a link to contact information on the home page (See Section 6.4 - Contact Information and User Feedback).

- Provide a link to legal information on the home page.

- Remember that all home pages should include links to the privacy statement, the SFA home page, and the U.S. Department of Education home page.

- Display logo and/or site title (agency name) prominently on the home page, as well as, all interior site pages.
- Provide ALT-text for company logo/site masthead.
- Provide viewing requirements on the home page.
- Keep the home page simple.
- Make the home page clear and appealing.

11.2. Menu-Based Home Page Design

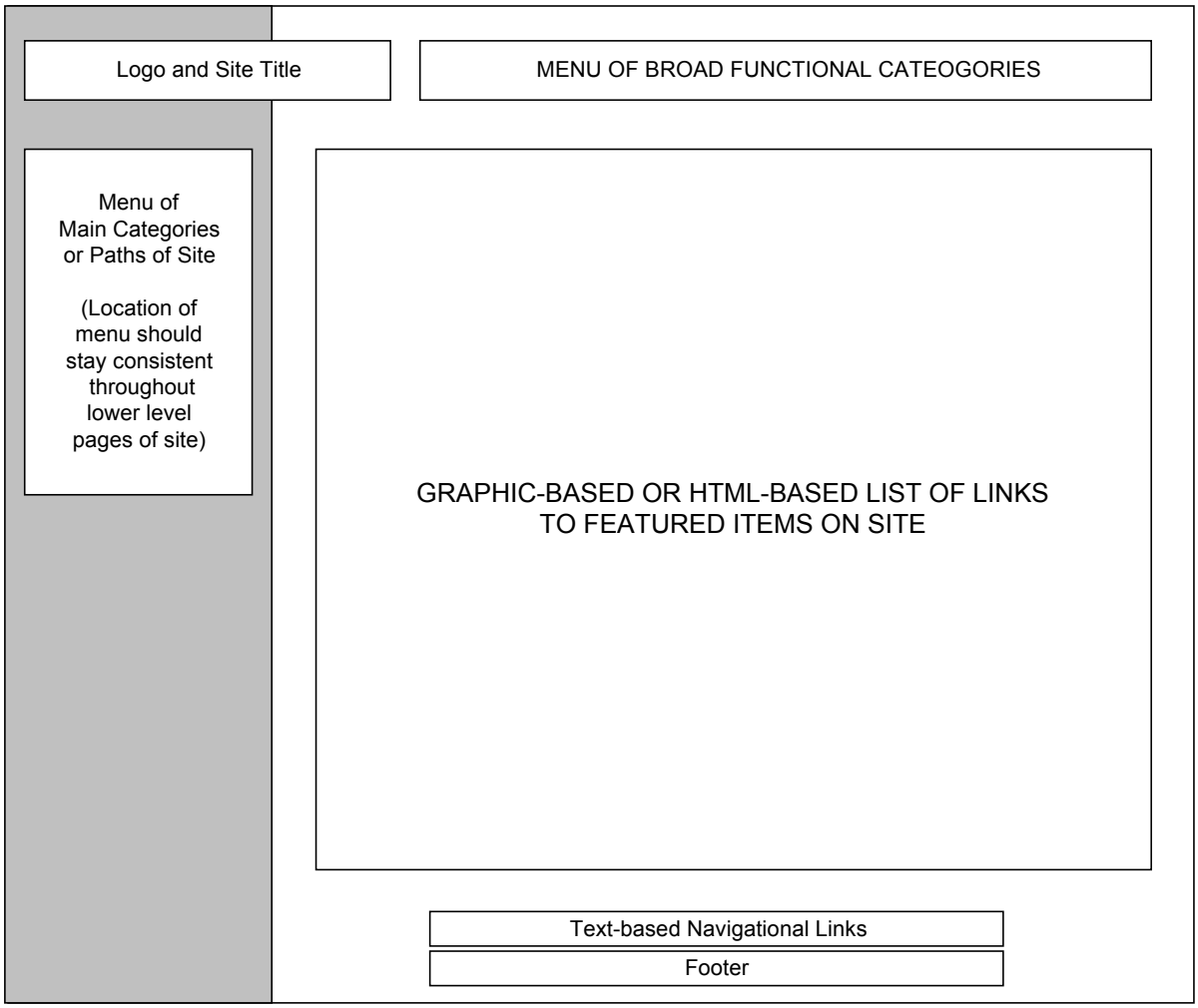
It is important that the home page imparts, both graphically and informationally, that it is the top level or entry point of the site. Menu-based home pages that offer lists of links and broad paths to the rest of the site are the most common type of home page. Larger web sites that are doorways for several audiences fare better with a menu-based home page design. The U.S. Department of Education's home page (http://www.ed.gov/) uses a menu-based home page design.

Menu-style home pages typically use either text-based HTML links or graphic image maps, or a combination of both. The menu style makes it possible to include a large number of site paths in a small amount of page real estate.

Category and path names should be strategically chosen and should be clear which audience or interest groups they serve, or should define areas of the site that contain specific information or achieve certain goals.

Links back to the home page should be available from every page in the site. The home page should promote a distinct design (while introducing common navigational and design elements) from the other pages. The overall organization and structure of the site should be conveyed by home page design. It should also clearly introduce the site's purpose and be segmented by what the key modules or sections and functions of the site are. There should be prominence (in terms of placement) given to the highly important, frequently accessed functions, modules or information, with consideration to audience's need.

Figure 11-1: Sample Layout for Menu-Based home pages



Note from Beacon: Add to Menu of Main Categories – Consider that the menu may also be dropped, or incorporated into an upper level navigation, when space constraints require it. For example, at the content level or for applications that may need the full width of page.

11.3. Design-Heavy Home Pages

Home pages that display strong graphic design characteristics sometimes serve mainly to attract new users to the Web site. These types of home pages are typically low on content and offer users links to the primary paths of the site.

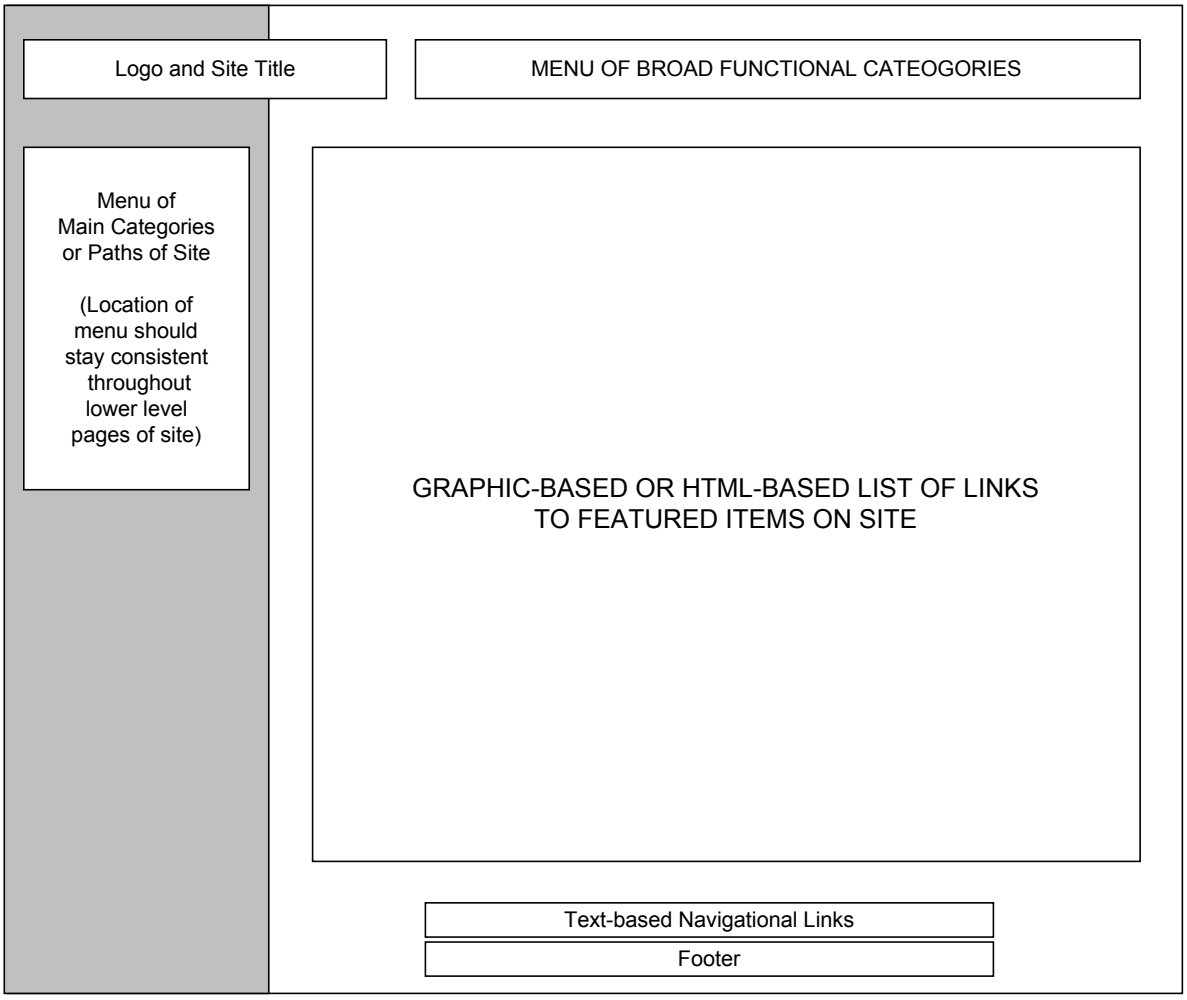
The design for this type of page should not only be aesthetic but functional, and particular care should be used to choose categories that are meaningful and clear to the user's point of view.

Graphic design on pages should be used to introduce and reinforce navigational and functional properties of the site – and/or content of particular value. An elegant but non-functional home page will annoy most regular or power users of the Web site.

An additional consideration should be given to the size and amount of graphics that are used on any page. The larger and more numerous the graphics, the longer it will take to load and display the page, increasing the chances that users will grow impatient and leave the site on their quest for more useful tools.

[Provide general guidelines for estimating the weight of a page, with all graphics, text, and links. Some products provide this functionality, such as Allaire HomeSite. Check our standard development tools for product-specific examples?]

Figure 11-2: Sample Layout for Design-based home pages



[PROVIDE SAMPLE LAYOUT. (This may be difficult, as one of the points of design-heavy pages is that we are removing the constraints of a typical layout, and no two design-centric applications look alike. The developers should have more freedom to create the layout of items on the page, while adhering to the rest of the GUI guidelines.)]

11.4. Portal Design Home Pages

The best-designed portal sites allow users to find the topics or information that they are looking for, and/or easily print or download what they find.

Graphics should be minimal, and content and menu structure must be carefully organized to support fast search and retrieval, easy downloading of files, and convenient printing options.

User contact time is typically brief on portal sites, particularly on the home page. Shorter “eyeball” time is better, as it implies that users are finding what they are looking for quickly.

Figure 11-3: Sample Layout for Portal Design home pages

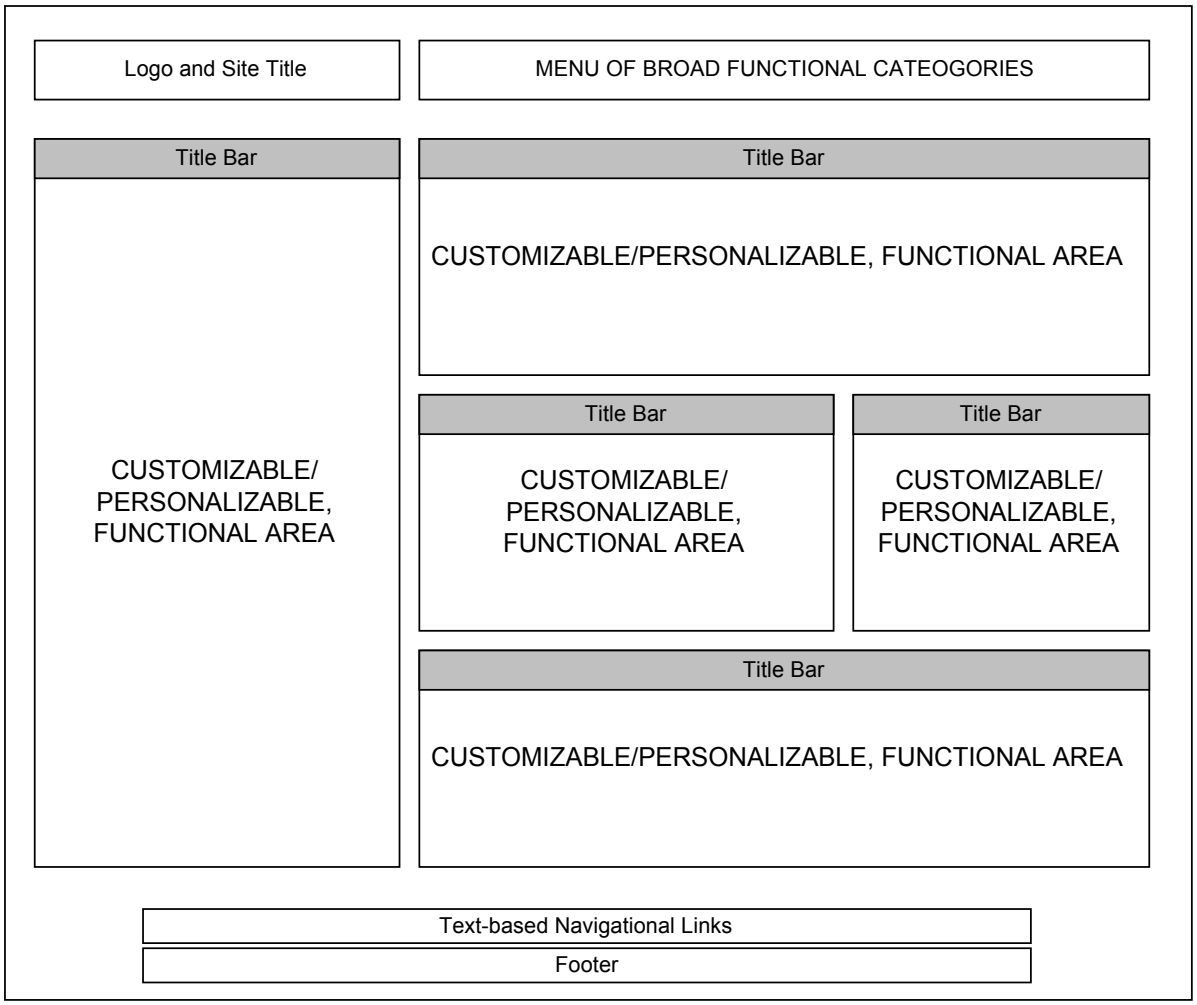
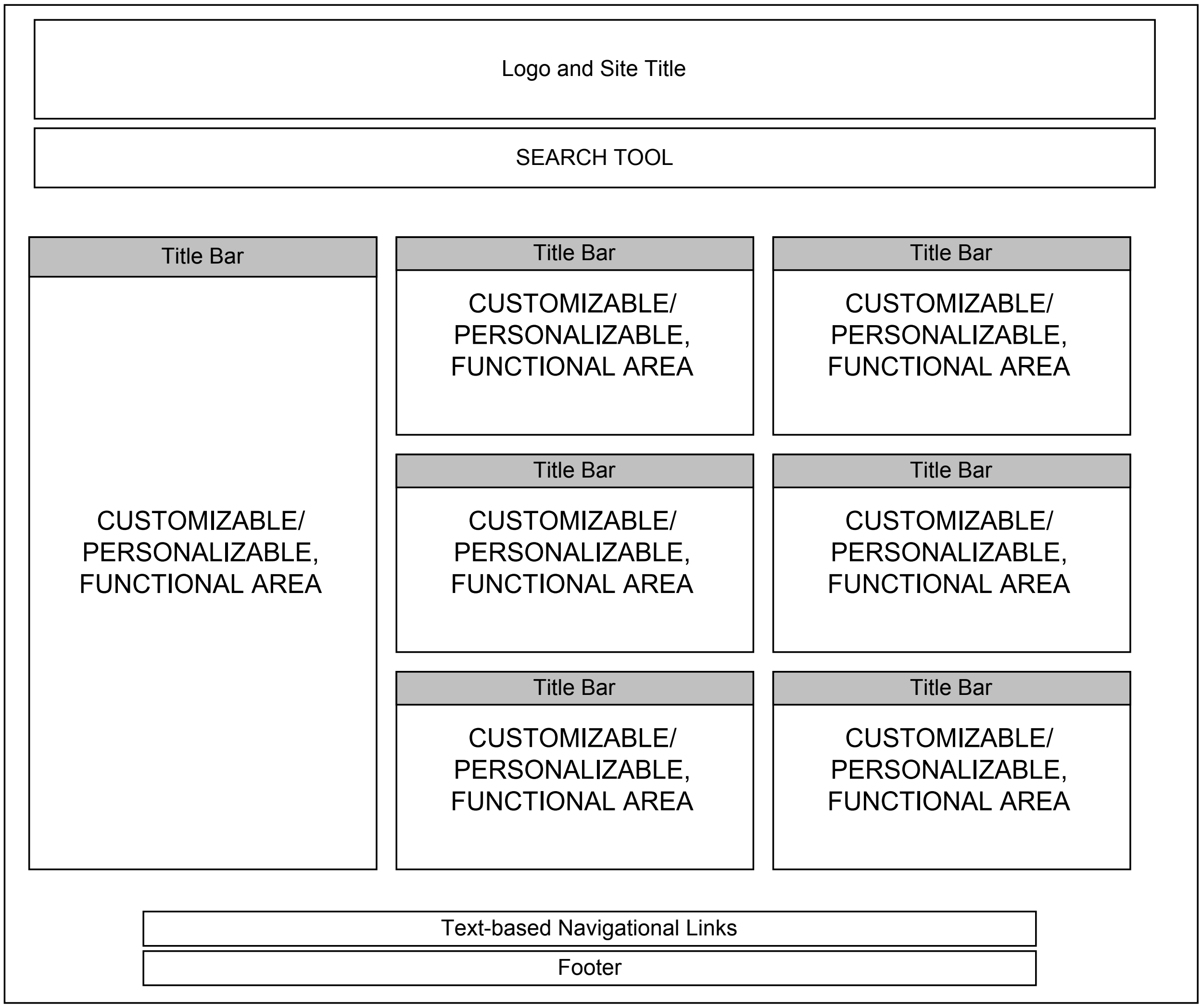


Figure 11-4: Another Sample Layout for Portal Design home pages



[The same problem applies to offering a "template" for portal designs – there are many approaches to layout of items on the page. I have provided two samples above. First of all, each functional area on a page may be moved around or customized by the user (function of personalization). Each user or group of users will likely have different pieces of functionality (different "views") that are associated with needs. Thus, portal-centric designs will differ according to the needs of the audience.]

11.5. News-Based Home Pages

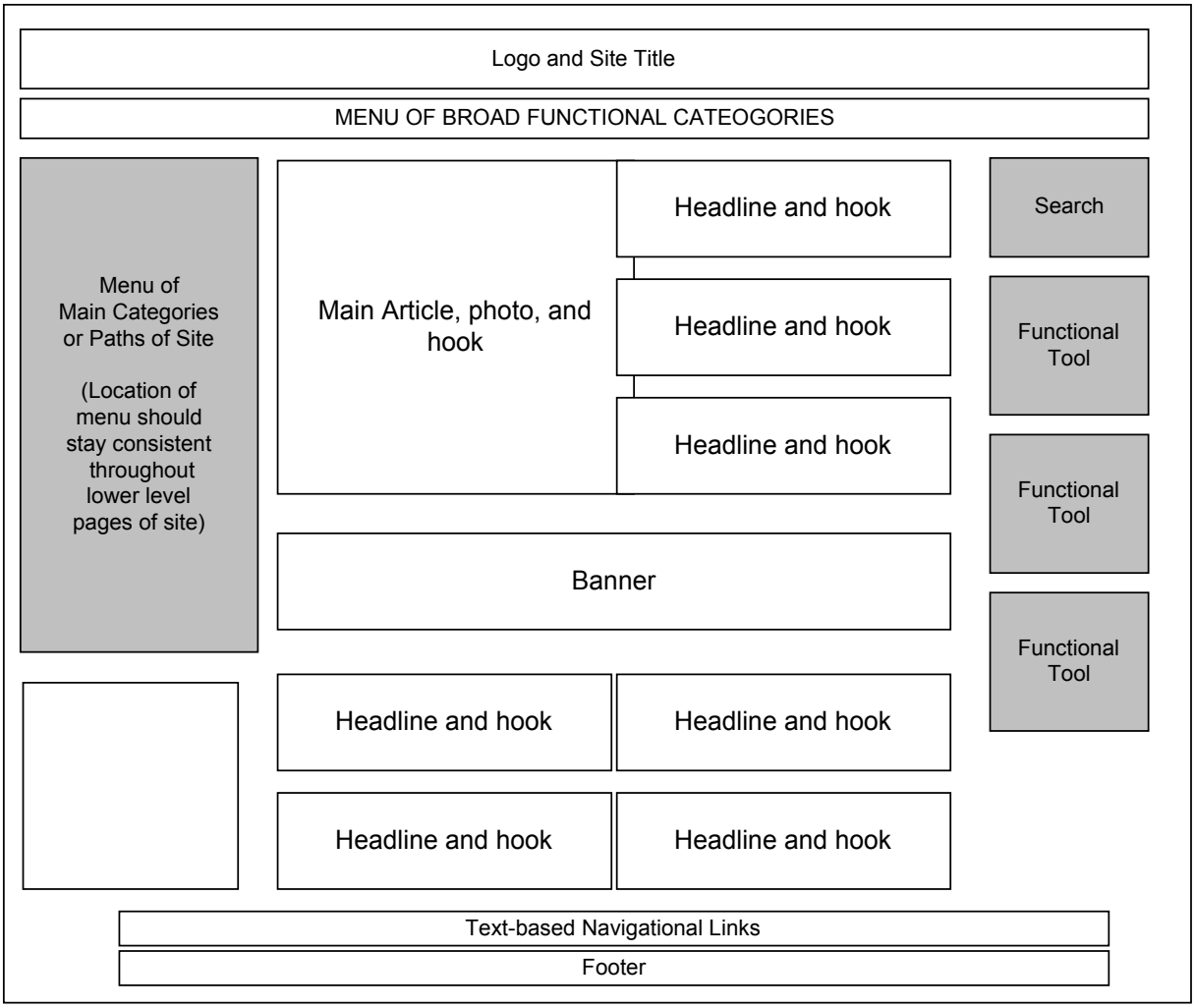
Web sites that are repositories of news-based information typically organize their home page into rectangular chunks of information (e.g. headlines and one or two sentences of block text) in order to

mimic the design of a newspaper. Sometimes this design strategy is incorporated into the overall portal design.

These types of web sites generally offer users access to quickly changing news, calendar events, and message alerts. Particular care should be given to keep the content on these sites extremely fresh, such as changing the content hourly, daily, or at most, weekly.

While the content itself should remain fresh, choose a standard framework for organizing information on the home page. The overall layout of the home page should remain consistent, or users will become confused on return visits.

Figure 11-5: Sample Layout for News-Based home pages



11.6. Lower-level Pages

The second and third level (or tier) pages of every Web site should use a standard layout grid that all internal pages of the site will share. As such, it is very important to design a strong, logical page grid that may be used consistently throughout a site for menu pages vs. content pages.

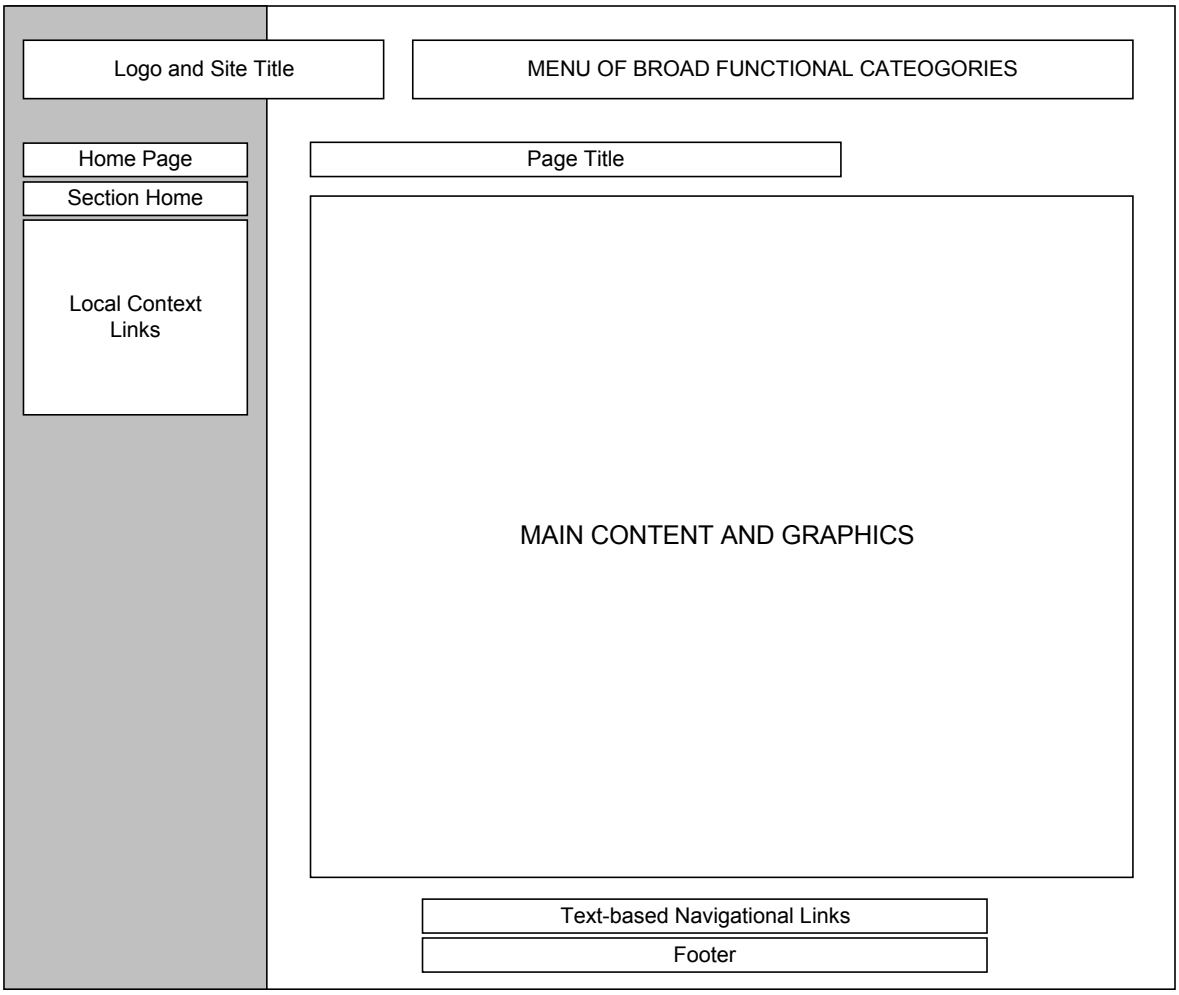
It is important to establish a regular, repeating pattern of carefully organized pages of text and graphics to help the user establish the location and organization of information and increase legibility.

Templates for establishing layout of lower-level pages work consistently well for the following types of functional pages:

- **Second and third level pages** (except for section or subsite home pages, which typically reflect the design template established by the Web site's home page)
- **Content-oriented pages** (Web pages that are text-heavy in the body of the page)
- **Form-based data-entry pages** (Web pages whose primary purpose is to collect information from the user, typically through the use of Web-based forms)

Generally, you should use the same approach to organize information on the page for all of the above styles of pages. Often, in fact, lower-level pages are content-oriented or form-based pages.

Figure 11-6: Sample Layout for Second-level Pages



- √ Provide a link to the home page on all interior site pages.

- √ Items that should be included on all interior site pages include:

  - Company name/logo
  - A link to the home page
  - Global navigation bar
  - Text-based navigation links at the bottom of all pages. Text-navigation should include, at a minimum, links to all top-level site areas.
  - Option to search or a link to the site search tool page

11.7. Forms

[Below is a suggested list of items to address in this section (outline taken from GUIGuide section on "Application Design – Forms."]

Choose buttons for web site forms [Need to write.]
Design internationally compatible forms [Need to write.]
Indicate protected areas of forms [Need to write.]
Initial placement of the cursor on forms [Need to write.]
Labeling input boxes on web forms [Need to write.]
Using an asterisk to show required fields [Need to write.]
Using tables to set up web forms [Need to write.]

11.8. Headers and Footers

11.8.1. Consistency of Headers and Footers

- The header is a set of information placed at the top of every Web page.

11.8.2. Consistency of Links in Headers and Footers

[Need to write.]

11.8.3. Content of Headers

- Every document should contain the following information in a standard header:

[Need to write.]

- A virtual include file may be used and referenced in the document to hold the header information. For example:

```
<!--#include virtual="/includes/header.html"-->
```

- Sample code for a standard header follows:

[Need to write.]

11.8.4. Content of Footers

- The footer is a set of information placed at the base of every page. Every document will contain the following information in a standard footer, separated from the body of the page by an appropriate amount of white space (usually two <BR> tags):

  - Link back to the top of the page (can be omitted if the page fits onto one screen)
  - Link to an appropriate home page (for the document, collection, project, program, organization, or Department) (Other page navigation should also link back to home and other pertinent points within the site.)
  - Links to additional documents in same section
  - Date that the document was last modified
  - Responsible engineer/contact person with person's name, email address (and mailto: link) [Wayne's Note: NEED TO DETERMINE HOW FEEDBACK WILL BE COLLECTED AND ROUTED – with generic mailboxes, mailboxes organized according to office, etc? Check content management standards to see if this is covered.]
  - "Student Financial Assistance" stamp

- A virtual include file may be used and referenced in the document to hold the footer information. For example:

```
<!--#include virtual="/includes/footer.html"-->
```

- Sample code for a standard footer follows:

  [THIS IS JUST A STARTING POINT. NEED TO MODIFY/VERIFY.]

```
<TR>
 <TD>
 <BR>
 <P ALIGN="center ">
 <FONT FACE="arial, helvetica" SIZE=1>
 <A HREF="page_name#top.html" CLASS="small">Top</A> |
 <A HREF="/index.html" CLASS="small">Section Home</A> |
 <A HREF="/folder_name/page_name1.html" CLASS=small>Content Item 1</A> |
 <A HREF="/folder_name/page_name2.html" CLASS=small>Content Item 2</A> |
 <A HREF="/folder_name/page_name3.html" CLASS=small>Content Item 3</A> |
 <A HREF="/folder_name/page_name4.html" CLASS=small>Content Item 4</A> |
 <A HREF="/folder_name/page_name5.html" CLASS=small>Content Item 5</A>
 </FONT>
<BR>
<FONT FACE="arial, helvetica" SIZE=-2 CLASS="pro">Last Modified:
(DATE)</FONT>
<BR>
```

```
<FONT FACE="arial, helvetica" SIZE=-2 CLASS="pro">Please send questions,
comments, and suggestions to <A HREF="mailto:SFAWebmaster@ed.gov">SFA
Webmaster</A></FONT>
<BR>
<FONT FACE="arial, helvetica" SIZE=-2 CLASS="pro">Student Financial
Assistance</FONT>
</P>
</TD>
</TR>
```

11.9. Cascading Style Sheets

[Below is a suggested list of items to address in this section (outline taken from GUIGuide section on "Application Design – Style Sheets."]

Avoiding overriding the user's style sheet
Avoiding relying on font effects from style sheets
Choosing linked vs. embedded style sheets
Consistency in class names of style sheets
Coordinating style sheets with existing HTML tags
Defining colors in style sheets
Designing style sheets
Placement of global style sheets
Providing scalability of style sheets
Specifying generic fonts in style sheets
Using as few style sheets as possible
Using negative margins in style sheets
Using non-global style sheets
Using style sheet fonts
Using the Style attribute in style sheets

A cascading style sheet may be used to apply all attributes to the page, and is included within the <HEAD> tags at the top of every HTML document. Cascading style sheets allow for changes to be made to the entire site quickly:

```
<LINK REL=STYLESHEET HREF="/sfa.css" TYPE="text/css">
```

Pages that use style sheets must still be readable when style sheets are not supported.

11.10. Window Design

[Below is a suggested list of items to address in this section (outline taken from GUIGuide section on "Application Design – Window Design."]

Arranging controls in windows
Avoiding group boxes for less than 7 items
Choosing the window size
Grouping and labeling information in windows
Making window navigation consistent/predictable
Navigating within a group box
Placement of labels in group boxes
Placing information in windows
Scrolling in windows
Separating content into short windows
Setting the margins of windows
Tabbing between controls in a window

11.11. Server Error Pages

The most commonly served error page is the infamous "404 File Not Found" page. A custom-designed and standard set of error pages that are consistent with the Web site's look and feel are recommended, particularly for web applications if they offer useful functionality, such as explain the error, suggest alternatives, or provide links to home page, site map, or search page.

12. Common Reusable Functions

[Wayne's note: Need to develop a small set of Java Beans that perform common functions (e.g. listbox, combobox, radio buttons, etc.). These may already exist in the Java AWT, but we should point developers to them as common reusable objects/beans for creating GUI components.]

12.1. Check Boxes

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Aligning check boxes
Disabling check boxes
Grouping related check boxes
Labeling check boxes
Maximum number of check boxes per page or window
Ordering check boxes
Placing labels for check boxes
Providing mnemonics for check boxes
Using check boxes to toggle options
Using check boxes vs. a multiple-selection list
Using check boxes vs. binary radio buttons
Using check boxes vs. data entry fields
Wording check box labels

12.2. Context Menus

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Avoiding accelerators in context menus
Disabling menu items in context menus
Making context menus consistent
Making Properties the last context menu item
Ordering menu items in context menus
Placing the default item first in context menus
Providing keyboard access to context menus
Providing mnemonics for context menu items
Using context menus for object-specific menus
Using ellipses (...) in context menus

Using separator bars in context menus

12.3. Custom Controls

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Avoiding custom controls
Designing custom controls

12.4. Data Entry Fields

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Accepting default values for data entry fields
Aligning data entry field labels
Aligning data entry fields
Appearance of enterable data entry fields
Appearance of non-enterable data entry fields
Avoiding auto-exit with data entry fields
Avoiding leading zeros in data entry fields
Avoiding long labels for data entry fields
Avoiding numbers and letters in data entry fields
Choosing default values for data entry fields
Completing input of data entry fields
Disabling data entry fields
Grouping related data entry fields
Labeling data entry fields
Making data entry fields accommodate pen-users
Making data entry fields case-blind
Making data entry fields similar lengths
Maximum number of different size data entry fields
Placing labels for data entry fields
Providing mnemonics for data entry fields
Showing number of spaces in data entry fields
Spacing between data entry fields and labels
Specifying measurement units for data entry fields
Using data entry fields for text input
Using data entry fields vs. list boxes
Wording data entry field labels

12.5. Dialog Boxes

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Avoiding concentric group boxes in dialog boxes
Grouping information in dialog boxes
Making mnemonics in a dialog box unique
Using expanding dialog boxes
Using modal dialog boxes

12.6. List Boxes

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Aligning list boxes
Avoiding scroll bars in multiple-selection lists
Disabling items in list boxes
Disabling list boxes
Disabling scroll bars in list boxes
Filtering list boxes with over 40 items
Grouping related list boxes
Labeling list boxes
Placing labels for list boxes
Providing mnemonics for list boxes
Showing items in a standard list box
Using a list box vs. radio buttons
Using combination list boxes
Using drop-down list boxes
Using extended list boxes
Using list boxes to display dynamic information
Using multiple-selection list boxes
Wording list box labels

12.7. Property Sheets

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Aligning push buttons on property sheets
Avoiding concentric group boxes on property sheets

Avoiding launching multiple property sheets
Closing property sheets that have no actions
Grouping information on tabs of property sheets
Making mnemonics in a property sheet unique
Making mnemonics on a property sheet unique
Making property sheets modeless
Naming property sheets
Placement of separator bars on property sheets
Placing icons on property sheets
Providing previews for property sheet settings
Rubber-banding controls on sizable property sheets
Selecting the sizing option for property sheets
Using one tab for simple property sheets
Using system fonts on property sheet tabs
Using What's This help on property sheets

12.8. Pull-Down Menus

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Avoiding multiple cascading pull-down menus
Avoiding repeating menu title in pull-down menus
Disabling menu items in pull-down menus
Disabling pull-down menus
Making accelerators for pull-down menus consistent
Minimum number of menu items in pull-down menus
Ordering menu items in pull-down menus
Placement of accelerators in pull-down menus
Providing mnemonics for pull-down menu items
Providing mnemonics for pull-down menu titles
Titling pull-down menus
Using accelerators for pull-down menu items
Using ellipses (...) in pull-down menus
Using pull-down menus to access commands
Using separator bars in pull-down menus

12.9. Push Buttons

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Aligning horizontally-placed push buttons
Aligning vertically-placed push buttons
Avoiding more than 2 sizes of push buttons
Avoiding multiple functions on push buttons
Choosing the default push button
Disabling push buttons
Grouping related push buttons
Labeling push buttons
Making push buttons a consistent shape
Making push buttons in a group box the same size
Maximum number of push buttons per page or window
Ordering push buttons
Placing labels for push buttons
Placing push buttons in a window
Positioning push buttons to show what they affect
Providing mnemonics for push buttons
Showing the default push button
Using 7 pixels of space between push buttons
Using Apply push buttons
Using Cancel push buttons
Using chevrons (>>) on push buttons
Using Close push buttons
Using ellipses (...) on push buttons
Using Help push buttons
Using OK push buttons
Using push buttons for immediate/critical actions
Using Reset push buttons
Wording push button labels

12.10. Radio Buttons

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on “Controls.”]

Aligning radio buttons
Disabling radio buttons
Grouping related radio buttons
Labeling radio buttons
Maximum number of radio buttons
Ordering radio buttons
Placing labels for radio buttons
Providing mnemonics for radio buttons

Using radio buttons for mutually exclusive choices
Wording radio button labels

12.11. Roll-Up Menus

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Sliders
Aligning sliders
Disabling sliders
Displaying the numerical position of sliders
Grouping related sliders
Labeling sliders
Placing labels for sliders
Presenting errors in slider values
Providing mnemonics for sliders
Showing the limits of sliders
Using sliders for continuous scales
Wording slider labels

12.12. Spin Boxes

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Aligning spin boxes
Disabling spin boxes
Grouping related spin boxes
Labeling spin boxes
Maximum number of spin box values
Placing labels for spin boxes
Presenting errors in spin box values
Providing dual entry of spin boxes
Providing mnemonics for spin boxes
Showing limits of spin boxes
Using spin boxes only for numerical values
Wording spin box labels

12.13. Tabs

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Avoiding concentric group boxes on tabs
Avoiding mnemonics for tab headings
Avoiding more than one row of tabs
Making mnemonics on tabs unique
Making the width of tabs consistent
Providing keyboard access to tabs
Sizing the side margins of tabs
Using tabs to group complex information

12.14. Text

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Capitalizing button names
Capitalizing icon names
Capitalizing labels and static text
Capitalizing menu items
Capitalizing menu titles
Capitalizing tab control labels
Using colons to end labels

12.15. Toolbars

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Avoiding overloading the toolbar
Avoiding scrolling toolbars
Providing an option to hide the toolbar
Sizing toolbar buttons

12.16. Tree View

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Providing visual clues for tree view
Using folders for tree view

12.17. Well Controls

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Labeling well controls
Placing labels for well controls
Providing mnemonics for well controls
Using well controls for graphical choices
Wording well control labels

12.18. Windows

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Controls."]

Allowing high frequency input in windows
Making menu bars one line
Making menu titles one word
Making mnemonics in a window unique
Placement of message bars
Placement of secondary windows
Providing an option to hide the status bar
Providing controls for child windows
Providing standard buttons in the title bar
Using cascading windows
Using message bars
Using non-interactive controls in the status bar
Using secondary windows to add functionality
Using tiled windows
Using workspaces

13.SFA HTML Coding Standards

13.1. Follow HTML standards

√ All SFA pages must support the HTML 4.0 standard. Although many Web pages do not include a reference to document type, you must include the following as the first line of the HTML document so that all SFA pages are identified as HTML 4.0 based:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

13.2. Major HTML Tags

√ Standard HTML tags are required for correct and consistent document presentation. The complete structure of a standard HTML 4.0 document is provided in Table 13-1 below.

Table 13-1: Standard HTML document

Tag Function	Use of standard HTML tags within an HTML document
Document Type	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
Opening HTML tag	<HTML>
Opening HEAD tag	<HEAD>
Title	<TITLE>Brief but Descriptive (and Unique) Title of Page</TITLE>
Closing HEAD tag	</HEAD>
Opening BODY tag	<BODY BGCOLOR="#ffffff" LEFTMARGIN="0" TOPMARGIN="0" MARGINHEIGHT="0" MARGINWIDTH="0" ALINK="#333399" VLINK="#333399" LINK="#333399" BACKGROUND="/images/background.gif">
The body tag contains the content displayed by your document.	Document Content.
Closing BODY tag	</BODY>
Closing HTML tag	</HTML>

- Indicate the natural language of the document by using the HTML <LANG> tag or XML <XML:LANG> tag.

- Use default, place holding characters in edit boxes and text areas for forms.

13.3. Make your HTML backwards compatible

[Need to write.]

13.4. Provide Meta Tags

Meta-data is data that describes data. Meta-data is used to describe the data in a document, but it is not part of the text of the document. In some Document Management Systems, meta-data is referred to as *properties*, *attributes*, or *custom variables*. If the information describes the document, then it is meta-data.

Meta-data provides an easy way to search for the document because the search can be narrowed down. Meta-data is the online way of classifying information, of putting documents into folders. Meta-data also allows a developer to identify a document as belonging to multiple categories without physically having to make a choice between one category or another, as you might have to do if you filed a piece of paper.

- All pages must contain the <META> keyword and description tags. An example of a standard <META> tags is provided below.

```
(NEED TO WRITE EXAMPLE)
```

13.5. Provide Web Page Titles

[Cover in this section:]
Avoiding articles in page titles
Avoiding jargon in page titles
Including document names in page titles
Making page or window titles informative
Number of words in page titles
Preparing the content of page titles
Providing page titles for all web pages
Using short descriptions for page titles

- Each page must contain a descriptive title (in the <TITLE> tag). This is the title that will appear when people search for information, when they create a “Favorites” listing in their browser, and

when they print the page.

- The title should be as short as possible but should be fully informative and specific (e.g., " Student Financial Assistance - Site Guide " is preferable to "Site Guide"). Although the title is often overlooked because it does not appear in the body of the document, it is important because it is frequently used to identify the document on hotlists, search result sets, and site indexes.

- The TITLE must provide enough contextual information to be used as a useful bookmark title. Do not assume that the user will be aware of the page's context, since the user might browse to the page directly from an external page or from a different section of the Web site.

- The title is not part of the text of the document, but is a property of the whole document. It may not contain anchors, paragraph marks, or highlighting.

- The title may be used to identify the node in a history list, to label the window displaying the node, etc. It is not normally displayed in the text of a document itself.

- The TITLE must not exceed 60 characters to avoid clipping when it is displayed in the titlebar of certain browsers.

✓ The <TITLE> tag is inserted between the opening and closing <HEAD> tags. The title will appear at the top of the respective browser and is also used by the browser to "bookmark" the page for future access. Additionally, some search engines use the <TITLE> as primary indexing point. The title tag is closed with </TITLE>.

```
<HEAD>
<TITLE>Student Financial Assistance - Site Guide</TITLE>
</HEAD>
```

13.6. Readability of HTML Code

13.6.1. Use Lowercase in HTML Tags

✓ HTML is case insensitive, but do not mix case with tags. **Use lowercase for all tags, name, and attributes.** The only exception is special characters. (Note: emerging standards prohibit the use of uppercase.)

13.6.2. Use Comment Tags

- The use of HTML comments to identify major and minor sections of code, components, functions, and navigational areas, and to identify the purpose, use, and outcomes of functions and

components, is highly recommended (e.g. **<!-- This is a comment -->**).

Δ Refrain from using comments to "comment out" a section containing other HTML tags because some browsers have trouble processing these areas, and unexpected errors may result.

13.6.3. Line Length

√ Do not create lines longer than **160 characters**. Although line length is unimportant to browsers, many editors handle the wrapping poorly making it hard to edit the HTML files.

13.6.4. Use of White Space

Δ Browsers ignore white space between lines. On a given line, multiple blank spaces are treated as one space and tabs are ignored. (None of the above apply between <PRE> and <PRE>, however.) Take advantage of white space by structuring and properly indenting your code so that your HTML code is more readable. However, bear in mind that tabs and carriage returns can add significant weight to file size. Excessive indentation is discouraged.

13.6.5. Use of Double Quotes in Attribute Values

Δ Although not all browsers require this, you should place double quotes around all the values of an attribute, not just when the value contains embedded spaces. (Technically, double quotes are only needed if you have a non-alphanumeric in the string, but it's best to always use them.) Note: Emerging standards require double quotes on all names and attributes.

```
<INPUT TYPE="submit" VALUE="Submit">
```

14.Accessibility Standards

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on “Accessibility.”]

Avoiding access barriers with posted documents
Designing help systems for the disabled
Hyperlinking audio/video captions with "CC" text
Hyperlinking photographs with "D" text
Providing accessibility alternatives for forms
Providing an access instruction page
Providing support for text browsers
Using "alt" tags with graphics for accessibility

Requirements for Accessible Software Design
Version 1.2 (May 31, 2000)
U.S. Department of Education - Office of the Chief Information Officer

The “Requirements for Accessible Software Design” document is available in alternate formats upon request by contacting the Alternate Format Center in the Mary E. Switzer Building at (202) 205-8113 or 260-9895.

14.1. Purpose

The Department of Education considers accessibility to information a priority for all employees and external customers, including individuals with disabilities. The Department has established these Requirements for Accessible Software Design in order to support its obligation, under Sections 504 and 508 of the Rehabilitation Act of 1973, 29 U.S.C. 794 and 794d. This is to ensure the accessibility of its programs and activities to individuals with disabilities, specifically its obligation to acquire accessible electronic and information technology. Therefore, when selecting computer hardware and software applications for use within the Department's computing environment, the Department will evaluate the hardware and software to determine its accessibility by users with disabilities.

The purpose of this document is to convey the accessibility needs of the Department to the developers and suppliers of computer applications. It addresses the minimum accessibility requirements software applications must meet in order to be used by all Department employees and customers. These requirements are offered to demonstrate the accessibility needs that must be considered when designing and developing software for the Department of Education. They address

proven techniques for the design of universally accessible software that can be used by individuals with or without a disability. Software considered for use by the Department must execute in the standard operating environment at the time of offering and be compatible with the accessibility tools, both hardware and software, in use by individuals with disabilities at the Department.

While a product that meets these requirements ensures minimum accessibility for individuals with disabilities, the Department of Education encourages software and technology developers to be creative and maximize their design of software that is universally accessible. More specific recommendations for how to design universally accessible software can be obtained from Links on our Web site: http://www.w3.org/WAI/

When the standards implementing section 508 of the Rehabilitation Act of 1973, as amended are released, they will supercede the functional specifications outlined below.

14.2. Functional Specifications

14.2.1. Keyboard Access

√ The software program must provide keyboard access to all functions of the application. All actions required or available by the program must be available with keystrokes.

√ Clear and precise instructions for the use of all keyboard functions shall be provided as part of the user documentation.

√ The software must have a logical tabbing order among fields, text boxes and focal points.

√ There must be a well-defined, programmatically exposed focal point that moves as the input focus changes so that assistive technology can track the focus.

√ The software shall not interfere with existing accessibility features built into the operating system. (Such as Sticky keys, Slow Keys, Repeat Keys, high contrast mode in Microsoft Windows)

Δ Avoid using timed responses if possible. If used, the ability to modify the timing parameter, by individual user, is necessary.

14.2.2. Icons

√ All icons and images shall have clear precise text labels, alt tags or tool tips included on the focus or provide a user-selected option of text-only buttons.

- The use of icons shall be consistent throughout the application.

- Provide pull-down menu equivalents for Icon functions (menu, tool and format bar).

14.2.3. Objects (Controls)

- Painted text is not accessible to all users. Use system text drawing tools so that screen reader software can interpret the text. The minimum information that must be provided to an assistive technology is text content, text input caret location and text attributes.

- Sufficient information must be provided about user interface object so that assistive technology can identify and communicate how the objects are used. (e.g., the object is an edit field or a check box which is checked).

14.2.4. Sounds

- Support the Sounds feature where built into the operating system. (Such as Microsoft Windows show sounds feature.)

- Provide an option to display a visual cue for all audio alerts. Minimize use of audio alerts in general.

- Provide an option to display all audio information in text format, either as closed captioning, a pop-up window or other means in parallel with the audio content.

- Provide an option to display all video information in text format, either as closed captioning, multi-media audio, a pop-up window, or other means in parallel with the video content.

14.2.5. Display

- Do not use Color-coding as the only means of conveying information or indicating an action. Always provide an alternative or parallel method that can be used by individuals who do not possess the ability to identify colors.

- The application must support user-defined color and high contrast settings for all user interface controls and client area content.

- Do not use patterned backgrounds behind text or important graphics or provide an option to turn off background patterns.

- Application colors should not be hard-coded, so that individual users can override application default fonts through the operating system, so printing and text displays can be changed.

√ Allow user adjustment of, or allow user to disable flashing, rotating or moving displays to the extent that it does not interfere with the purpose of the application. Minimize use of moving display elements.

14.2.6. Field Labeling

√ Position the descriptions or labels for data fields immediately next to the field, so that it is easy for screen reading software, used by individuals that are blind, to associate the labels with the corresponding fields. The preferred position would be flush against the right side of the field. Associate labels clearly and programmatically with controls and objects.

14.2.7. Documentation

√ Provide all manuals and documentation in electronic format as an ASCII text or HTML file. This should include text descriptions of any charts and/or graphs or pictures or graphics of any nature. This is done to ensure that the information presented in charts or graphs is available to screen readers and/or in Braille versions of the text.

√ Any reports, forms, or documents that the application generates must be available in a "print to ASCII file" format.

14.2.8. Common Accessibility Aids

The Department of Education commonly uses, but is not limited to the following assistive technology aids:

- Henter Joyce JAWS
- AiSquared Zoom Text for Windows
- Dragon Systems, Inc. Dragon NaturallySpeaking Voice Recognition Software
- GUS! Word Prediction Software
- NXI Communications, NexTalk for Networks

15. Usability Standards

[NOTE: One of the Intranet Content Management documents is on usability standards. That document, in addition to the suggested topics below, could be integrated to form this section.]

15.1. Feedback

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Usability."]

Avoiding humor in error messages
Avoiding OK buttons in error messages
Making message bar messages concise
Ordering information in an error message
Placement of error messages
Using a long wait indicator for a known duration
Using a short wait indicator
Using feedback forms on web sites
Using questions in messages
Writing messages

15.2. Help Systems

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Usability."]

Allowing the user to bookmark help topics
Avoiding deeply nested help topics
Avoiding multiple help windows
Avoiding relying on on-line help
Evoking What's This help systems
Making help windows modeless
Making task help "always on top"
Providing the user with access to on-line help
Providing various levels of help
Using browse buttons in help systems
Using hint mode help systems
Using jumps in help systems
Using pop-ups in help systems
Using search functions in help systems

Using task-based help systems
Using the message bar to provide help
Using What's This help systems
Writing help topics

15.3. Tool Tips

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Usability."]

Ensuring that the user wants to see the tool tip
Providing an option to turn tool tips off
Showing the tool tip in a small window
Showing tool tips on mouseovers
Supplementing toolbar buttons with tool tips
Using a label instead of a tool tip
Writing the text for tool tips

15.4. Wizards

[Below is a suggested list of items to address or include in this section (outline taken from GUIGuide section on "Usability."]

Avoiding auto-advancing wizard pages
Avoiding leaving the wizard to complete the task
Avoiding technical terminology in wizards
Identifying the purpose of the wizard in the title
Indicating the wizard is on the last page
Limiting secondary windows in wizards
Limiting the number of choices in a wizard
Making wizards provide positive results
Phrasing questions in wizards
Providing a description on the first wizard page
Showing that the wizard graphic is not interactive
Showing the wizard's purpose with a graphic
Using a single secondary window for wizards
Using the Finish button in wizards
Warning about leaving an unfinished wizard
Writing the content of wizards

16. Additional Resources

This guide reflects the experience of many developers as well as documented "best practices" at many companies.

16.1. Style and User Interface Resources

Developers who are interested in additional information about style and other user interface issues should refer to the Internet resources listed in Table 16-1 below.

Table 16-1: Style and User Interface Resources

Style and User Interface Resources on the World Wide Web
Apple Web Design Guide http://applenet.apple.com/hi/Web/Web.html
Dave Siegel's Tips for Web Writers and Designers http://www.dsiegel.com/tips/
Designing HTML Pages to Increase Accessibility to Users With Disabilities http://www.trace.wisc.edu/HTMLgide/htmlgide.html
Gif89a http://www.cnet.com/Content/Features/Techno/Gif89
Graphics Research Lab Notes on Writing for Interactive Media http://www.electric-pages.com/notes/n1.htm
Matterform Media Qbullets http://www.matterform.com/qbullets/index.html
National Center for Supercomputing Applications (NCSA) Review of Style Guides http://www.ncsa.uiuc.edu/edu/trg/styleguide/index.html
Style Guide http://www.cl.cam.ac.uk/users/gdr11/style-guide.html
Sun Guide to Web Style http://www.sun.com/styleguide/
The Intranet Journal Design Tools http://www.Intranetjournal.com/design.html
Usability Testing of WWW Designs http://www.sun.com:80/sun-on-net/uidesign/usabilitytest.html

Web Pages That Suck
http://www.Webpagesthatsuck.com

Webmaster Magazine Writing Style Resources
http://www.cio.com/WebMaster/style.html

What is Good Hypertext Writing
http://kbs.cs.tu-berlin.de/~jutta/ht/writing.html

World Wide Web Consortium (W3C) Style Guide for Online Hypertext
http://www.w3.org/pub/WWW/Provider/Style/Overview.html

Yale University School of Medicine Center for Advanced Instructional Media Style Manual

- **First Edition:**
  http://info.med.yale.edu/caim/StyleManual_Top.HTML
- **Second Edition (Requires JavaScript):**
  http://info.med.yale.edu/caim/manual/contents.html

16.2. Accessibility Resources

Developers who are interested in additional information about style and other user interface issues should refer to the Internet resources listed in Table 16-2 below.

Table 16-2: Accessibility Resources

Accessibility Resources on the World Wide Web
Requirements for accessible software design http://gcs.ed.gov/coninfo/clibrary/software.htm **Rehabilitation Act of 1973** http://trace.wisc.edu/gofr_web/rehabact.html • **Section 508:** Electronic and information technology accessibility guidelines (updated with 1998 Amendments) http://www.itpolicy.gsa.gov/cita/section508.htm **Federal IT Accessibility Initiative (FITAI)** Web Accessibility Training Information and Registration http://www.itpolicy.gsa.gov/cita/training.html **W3C Web Content Accessibility Guidelines 1.0** http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/ **Curriculum for Web Content Accessibility Guidelines 1.0** http://www.w3.org/WAI/wcag-curric/ **Center for IT Accommodation (CITA)** http://www.itpolicy.gsa.gov/cita/

17. References

ED World Wide Web Policy and Procedures - Technical Standards and Guidelines
U.S. Department of Education
Office of the Chief Information Officer
http://www.ed.gov/internal/wwwstds/technical.html

GUIGuide ™
Classis Systems Solutions, Inc.
http://www.guiguide.com/

Intranet Content Management Style Guide Version 4.18.00 (draft)
U.S. Department of Education
Student Financial Assistance

Project EASI-ED System-Wide Design Standards (SWDS) Document (May 24, 1999)
U.S. Department of Education
Student Financial Assistance
http://easi.ed.gov/inside_projecteasi/HTML/techinfo/swds.html

Requirements for Accessible Software Design Version 1.2 (May 31, 2000)
U.S. Department of Education
Office of the Chief Information Officer
http://gcs.ed.gov/CONINFO/clibrary/software.htm

SCO Visual Tcl Programmer's Guide and Reference
Appendix D, User interface style conventions
http://compy.ww.tu-berlin.de/VTcl_DOC/VTCLG/vtclgN.style_goodui.html

Web Style Guide
Patrick J. Lynch and Sarah Horton
Yale University Press
http://info.med.yale.edu/caim/manual/